NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

REALISTIC INTERFACE AND CONTROL OF A VIRTUAL SUBMARINE MODEL IN NPSNET

by

Jeffrey A. Halvorson

March 1997

Thesis Advisor:	Michael Zyda
Thesis Co-Advisors:	John Falby
	Don Brutzman

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE March 1997	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE
REALISTIC INTERFACE AND CONTROL OF A VIRTUAL SUBMARINE MODEL IN NPSNET

5. FUNDING NUMBERS

6. AUTHOR(S)
Halvorson, Jeffrey A.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School Code
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/ MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/ MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the United States Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The current NPSNET submarine simulator interface is not a viable training tool because it utilizes a control panel which runs as a separate process and includes three separate tabs, one each for the Officer of the Deck, Helm, and Weapons Officer. Besides lacking immersion qualities, most of the control icons and prototypes are not functional.

Our approach was to mount human entities to the submarine that can control and maneuver the submarine entity by interacting with various objects onboard the submarine. These human entities represent key members of the submarine control party including the Officer of the Deck, Diving Officer of the Watch, Chief of the Watch, Helmsman/Planesman, and a second Planesman. The submarine model was improved by building a 3D Control Room and adding manipulable visual cues to represent an actual submarine control room.

As a result of this work, a group of human entities can operate a submarine in NPSNET, acting together as a watch team and maneuvering the submarine through the virtual environment. Realism has been improved by immersing a user into the virtual environment as a virtual human entity.

14. SUBJECT TERMS	15. NUMBER OF PAGES
NPSNET, DIS, real-time, 3D, visual simulation, network, distributed, interactive, virtual world, Potentially Visible Sets (PVS), Voice Recognition, mounting human entities, control panels	92
	16. PRICE CODE N/A

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 | Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

REALISTIC INTERFACE AND CONTROL OF A VIRTUAL SUBMARINE MODEL IN NPSNET

Jeffrey A. Halvorson
Lieutenant, United States Navy
B.S.M.E. University of Idaho, 1990

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 1997

Author: ______________________
Jeffrey A. Halvorson

Approved by: ______________________
Michael J. Zyda, Thesis Advisor

____
John S. Falby, Thesis Co-Advisor

____
Don Brutzman, Thesis Co-Advisor

____
Ted Lewis, Chair,
Department of Computer Science

ABSTRACT

The current NPSNET submarine simulator interface is not a viable training tool because it utilizes a control panel which runs as a separate process and includes three separate tabs, one each for the Officer of the Deck, Helm, and Weapons Officer. Besides lacking immersion qualities, most of the control icons and prototypes are not functional.

Our approach is to mount human entities to the submarine that can control and maneuver the submarine entity by interacting with various objects onboard the submarine. These human entities represent key members of the submarine control party including the Officer of the Deck, Diving Officer of the Watch, Chief of the Watch, Helmsman/ Planesman, and a second Planesman. The submarine model was improved by building a 3D Control Room and adding manipulable visual cues to represent an actual submarine control room.

As a result of this work, a group of human entities can mount a submarine in NPSNET, acting together as a watch team and maneuvering the submarine through the virtual environment. Realism has been provided by immersing a user into the virtual environment as a virtual human entity.

TABLE OF CONTENTS

I. INTRODUCTION 1

- A. BACKGROUND 1
  - 1. NPSNET 1
  - 2. NPSNET Submarine Simulator 2
  - 3. NPSNET Ship Simulator 3
- B. MOTIVATION 4
  - 1. Training Members of the Control Party 5
  - 2. Limited Steaming Time 6
- C. OBJECTIVES 6
  - 1. The Control Party 6
    - a. Officer of the Deck (OOD) 7
    - b. Diving Officer of The Watch (DOOW) 7
    - c. Chief of the Watch (COW) 7
    - d. Helmsman 8
    - e. Planesman 8
  - 2. Hydrodynamics Model 8
- D. THESIS OUTLINE 8

II. RELATED WORK 11

- A. INTRODUCTION 11
- B. NPSNET 11
- C. NPSNET SUBMARINE SIMULATOR 13
- D. NPSNET SHIP SIMULATOR 16
- E. VESUB 18
- F. SUMMARY 22

III. SYSTEM OVERVIEW 23

- A. INTRODUCTION 23
- B. DESIGN PHILOSOPHY 23

C. NPSNET SUBMERSIBLE ENTITIES 25
D. MULTIGEN OPEN FLIGHT MODEL 27
E. REMOTE CONTROL PANEL 28
F. NPSNET HUMAN ENTITIES 28
G. HIGH-RESOLUTION NETWORK 29
H. SUMMARY 30
IV. SINGLE USER INTERFACE 31
A. INTRODUCTION 31
B. KEYBOARD 31
C. CONTROL PANEL 32
D. SUMMARY 36
V. MOUNTING OF HUMAN ENTITIES 39
A. INTRODUCTION 39
B. CHANGING THE MULTIGEN OPEN FLIGHT MODEL 40
1. Conning Station 41
2. Helm Station 43
3. Dive Planesman Station 44
4. Diving Officer of the Watch station 44
5. Chief of the Watch Station 45
C. MOUNTING ALGORITHM 46
1. Maintaining Correct Position of Mounted Human Entities 47
2. Maintaining Correct Position of Networked Ghost Human Entities 49
D. HIGH RESOLUTION NETWORK 49
E. SUMMARY 51
VI. CONCLUSIONS AND RECOMMENDATIONS 53
A. RESULTS 53
1. MultiGen OpenFlight Submarine Model 53
2. The Single User Interface 54
3. The Watch Team Interface 54

4. Communications 55
5. Working with NPSNET 55
B. RECOMMENDATIONS FOR FUTURE WORK 55
APPENDIX A. MODEL HIERARCHY 57
APPENDIX B. VIDEO DEMONSTRATION 71
A. INTRODUCTION 71
B. SINGLE USER INTERFACE 71
C. MOUNTED HUMAN ENTITY INTERFACE 71
APPENDIX C. OBTAINING NPSNET SOFTWARE AND DOCUMENTS.................. 73
A. OBTAINING NPSNET SOFTWARE 73
B. OBTAINING NSPNET DOCUMENTS 73
LIST OF REFERENCES 75
INITIAL DISTRIBUTION LIST 77

LIST OF FIGURES

Figure 1. NPSNET Virtual Submarine 2

Figure 2. NPSNET Ship (Antares) with mounted human entity 3

Figure 3. Fuel Oil Fire in Engine Room Lower Level..4

Figure 4. NPSNET Coordinate Axis Convention 12

Figure 5. Helm Tab of Submarine Control Panel 15

Figure 6. OOD Tab of Submarine Control Panel 16

Figure 7. Weaps Tab of Submarine Control Panel 16

Figure 8. Ship Simulator Control Panel...17

Figure 9. Artist's Representation of VESUB System [VESUB 97] 19

Figure 10. HMD Headset in VESUB [VESUB 97] 20

Figure 11. HMD Harbor View [VESUB 97] 20

Figure 12. HMD Bridge View [VESUB 97] 21

Figure 13. NPSNET Class Hierarchy for Vehicles 25

Figure 14. NPS AUV Hydrodynamics Model Class Hierarchy 26

Figure 15. Submarine Control Panel 33

Figure 16. View from Port Periscope at Periscope Depth 34

Figure 17. IDU Packet Header... 35

Figure 18. IDU Packet Structure from NPSNET 35

Figure 19. IDU Packet Structure from Control Panel 36

Figure 20. Submarine Control Room 41

Figure 21. Conning Station of Submarine Control Room 42

Figure 22. Port Periscope Housing 42
Figure 23. Helm Station of Submarine Control Room 43
Figure 24. Dive Planesman Station of Submarine Control Room.................... 44
Figure 25. Chief of the Watch Station 45
Figure 26. OOD's view of Control Room from Conning Station 46
Figure 27. Class Hierarchy for Human Entities in NPSNET 47
Figure 28. Calculating Mounted Human Entity Global Position 48
Figure 29. Upper Level of Submarine Model Hierarchy 58
Figure 30. Hull Group Node and below of Submarine Model Hierarchy 59
Figure 31. Control Group Node and below of Submarine Model Hierarchy 59
Figure 32. Helm Group Node and below of Submarine Model Hierarchy 60
Figure 33. Dive Stick Group Node and below of Submarine Model Hierarchy 61
Figure 34. Sail and Hatch Group Nodes and below of Submarine Model Hierarchy 62
Figure 35. Cleats Group Node and below of Submarine Model Hierarchy.................... 62
Figure 36. Port Scope Assy Node and below of Submarine Model Hierarchy 63
Figure 37. Stbd Scope Group Node and below of Submarine Model Hierarchy 64
Figure 38. Fwd Planes Group Node and below of Submarine Model Hierarchy 65
Figure 39. Rudder Group node and below of Submarine Model Hierarchy 65
Figure 40. Stern Area Group Node and below of Submarine Model Hierarchy 66

LIST OF TABLES

Table1: HIRESNET Data PDU Structure [Stewart 96] 50

Table2: Summary of Group Node Types 67

Table3: Summary of PVS Cells 68

I. INTRODUCTION

A. BACKGROUND

1. NPSNET

The Naval Postgraduate School Networked Virtual Environment (NPSNET) is an interactive distributed virtual simulation system incorporating models of several types of military entities including helicopters, fixed wing aircraft, humans, surface ships and submersible vehicles. The development of this system began in 1990 with the efforts of numerous researchers and students. NPSNET's functionality and capabilities have improved with each generation of software, networking technology and graphics capabilities. [Zyda 94]

Currently, NPSNET is in its fourth major version (NPSNET-IV) with a fifth major revision in progress. Included with this system is a complementary suite of software applications such as network management tools and interface management for keyboard input, flight control system joystick and throttle, control panels and basic voice recognition. NPSNET uses the Distributed Interactive Systems (DIS) Protocol for networked communications [Barham 94]. Currently DIS version 2.0.4 is being used by NPSNET.

Several software applications currently exist that serve to simulate actual military training scenarios. Current applications support entities such as ships, submarines, tanks, fighter planes and infantry soldiers. By utilizing virtual simulations the costs and possible hazards of training on "real" equipment can be avoided, and ideally the training can be just as meaningful.

A unique feature of NPSNET is that applications are networked on multiple computers and offer real-time interactions among the software applications. With the DIS protocol, real-time three-dimensional (3D) contacts can exchange data with entities located at other workstations possibly thousands of miles away.

This thesis is an effort to add to the ongoing work to improve NPSNET. Specifically, the interface has been improved by mounting human entities to the submarine and providing them the capability to manipulate various objects aboard the submarine.

2. NPSNET Submarine Simulator

The NPSNET Submarine Simulator provides a submersible vehicle whose motion is determined by a real-time hydrodynamics model such that it moves through the virtual world according to realistic physically based models [Bacon 95]. With the use of Distributed Interactive Simulation (DIS) multiple ships, submarines and other entities can participate in a war-fighting training scenario in the same virtual world. Figure 1 depicts the NPSNET submarine surfaced in the virtual environment.

Multiple users can control the same submarine entity utilizing a multi-controller protocol with inputs from separate control panels. This model operates in an environment which includes an ocean carpet to simulate the environment below the ocean surface [Bacon 95]. Previously, when an entity dived below the ocean's surface, there was nothing there, hindering the reality of the simulation.

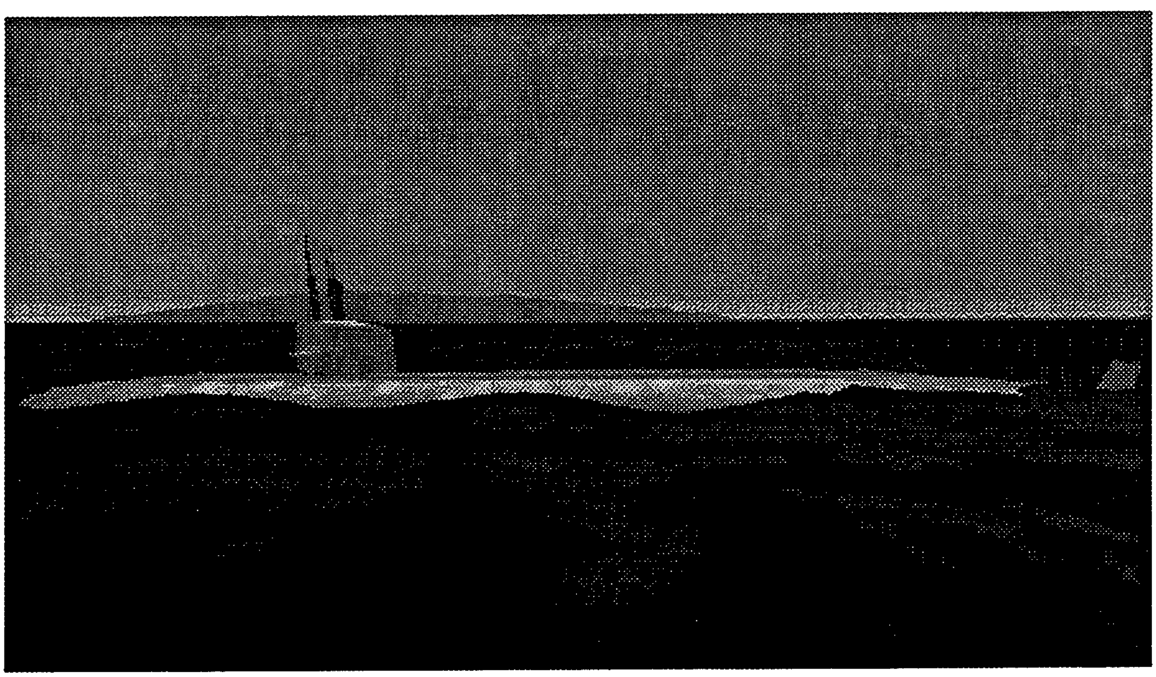

Figure 1: NPSNET Virtual Submarine

This thesis utilizes the basic submarine model from the submarine simulator. In addition the hydrodynamic equations of motion from the submarine simulator are used.

3. NPSNET Ship Simulator

NPSNET provides the ability to simulate various types of surface ships. One such model is the Antares. The Antares is a model of a proposed roll-on/roll-off commercial ship which was built for Naval Sea Systems (NAVSEA) by Advance Marine Vehicles [Obyrne 95]. Human entities can mount the Antares ship model, move about, and interact with various internal objects (e.g. doors, valves, etc.) while the ship is moving through the virtual environment [Stewart 96]. Figure 2 show the Antares ship model with a mounted human entity positioned on the boarding ramp.

This system allows users to participate in simulated casualties including fires and steam leaks. Mounted human entities can manipulate objects such as a fire hose nozzle, shutoff valves and ventilation switches to combat the casualty. Figure 3 depicts a mounted human entity fighting a fire onboard the Antares.

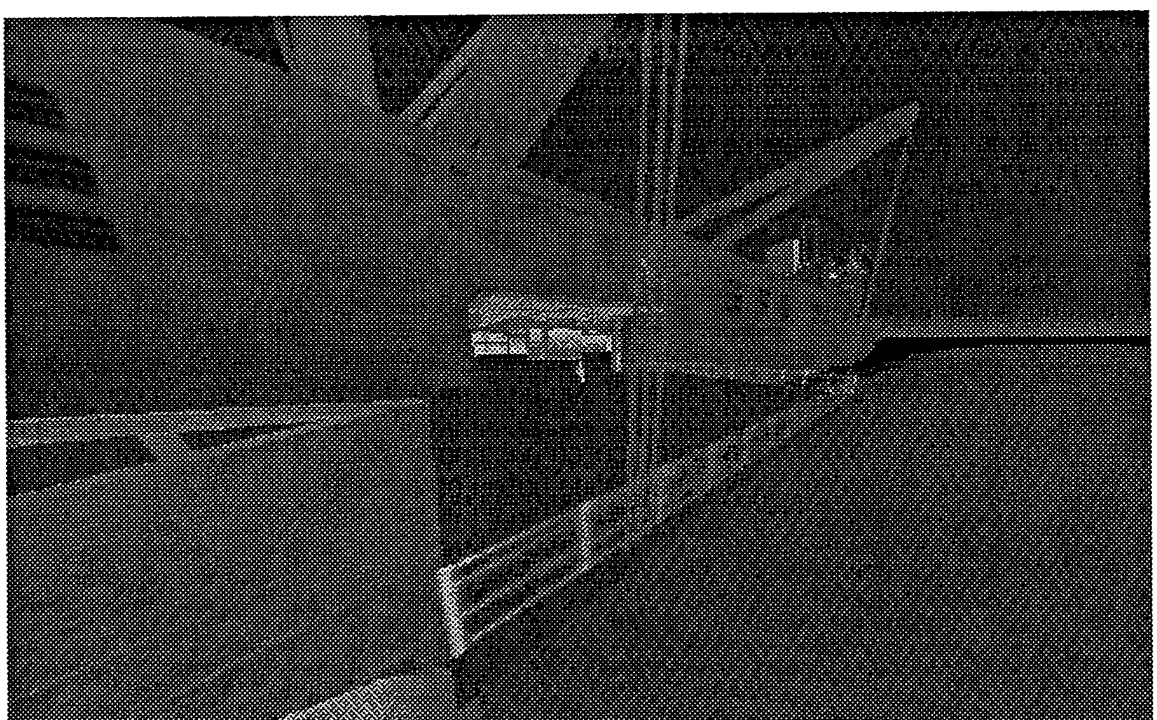

Figure 2: NPSNET Ship (Antares) with mounted human entity

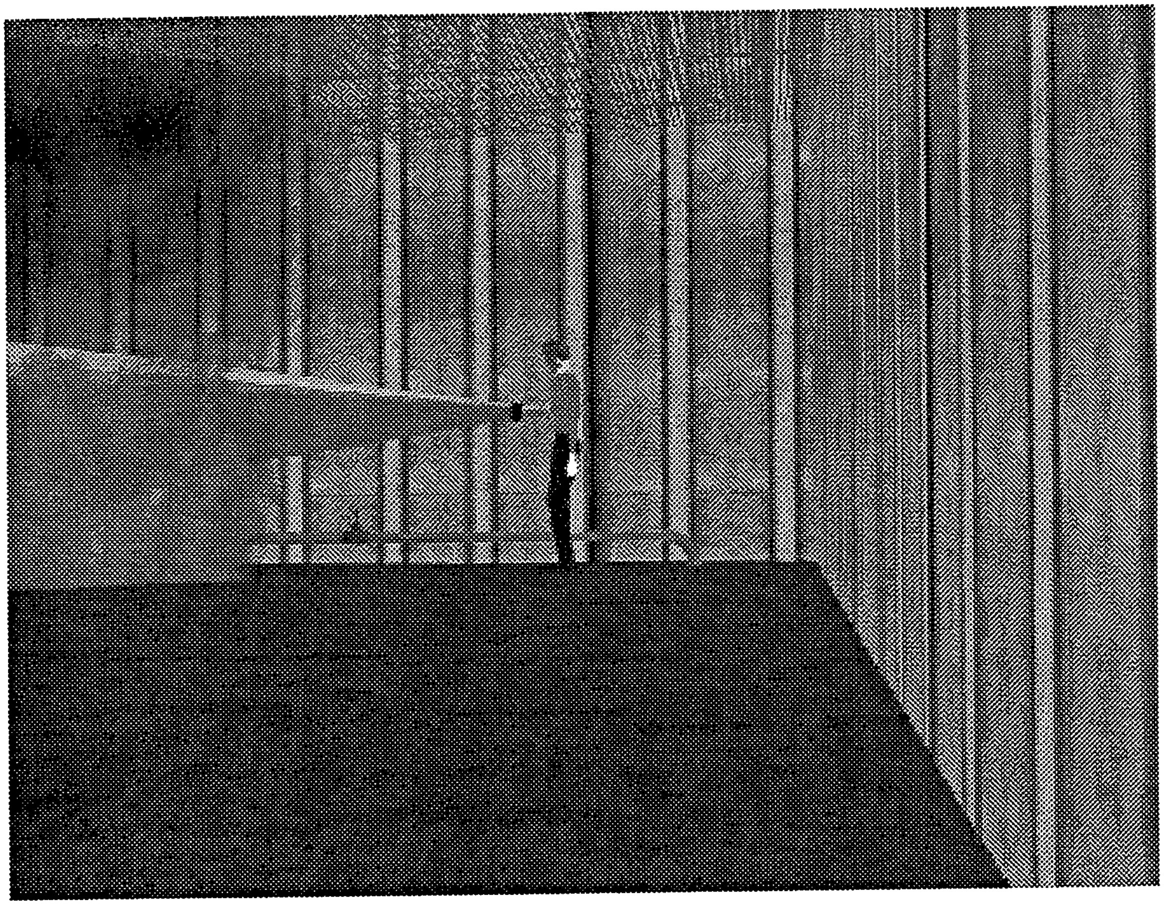

Figure 3: Fuel Oil Fire in Engine Room Lower Level

Some of the techniques used in the development of the ship simulator are pertinent to this research. For example, since this research work involves the mounting of human entities to a submarine, techniques used in the ship simulator to mount human entities are very useful. Also, techniques used to manipulate various objects onboard the ship are pertinent to similar work on the submarine simulator.

B. MOTIVATION

Simulation of battlefield events in a virtual environment offers several advantages over "real" training. Simulators can systematically train for a wide range of possible scenarios without the high cost and risk of actual flight time in an Air Force jet or cruise time in a Navy submarine [Pioch 95]. Also, injuries to inexperienced personnel can be avoided by

first gaining proficiency on a computer simulator. By gaining a higher level of proficiency before operating potentially hazardous equipment, injuries due to improper operation of equipment by inexperienced sailors can be avoided to a larger extent. Various entities can interact in a scenario without ever having to fuel up a tank or plane, load weapons, or arrange for tug support to get underway. These expenses can be saved while still achieving the benefit of realistic training in key areas such as ship handling and response to casualties.

1. Training Members of the Control Party

The development of a multi-controller protocol [Bacon 95] provided a means for multiple workstations to control the same entity (specifically a submarine) simultaneously. Each workstation can operate either one of three separate control panel tabs for the Officer of the Deck, Helm, or Weapons Officer. Unfortunately, this prototype interface is limited not only in its immersive potential but also in its current functionality. The user is detached from the virtual environment, manipulating buttons on a control panel which is not a part of the virtual environment. Functionality is almost non existent as most of the control panel buttons do nothing to affect the simulation.

To provide a realistic training scenario, an interface is needed that allows various Control Party members to communicate with each other, and to carry out actions associated with the operation of the ship's control surfaces and machinery. A natural solution is to mount or place human entities on the submarine entity which represent watchstanders of the control party. These Control Party members are the members of the watch section concerned with the safe navigation of the ship through the water. Key members include the Officer of the Deck (OOD), Diving Officer of the Watch (DOOW), Chief of the Watch (COW), a Helmsman/Planesman to control the rudder and one set of planes, and another planesman to control the other set of planes. Each of these entities therefore can control inputs to the submarine model from their respective watch station. For example, the Helmsman inputs propulsion orders to maneuvering via a simulated Engine Order Telegraph (EOT).

2. Limited Steaming Time

The defense draw down of the last decade has greatly reduced the number of fast attack and ballistic missile submarines. As such, there is less steaming time available to train junior officers in the art and science of ship handling. Similarly, less steaming time is available to train other members of the ship's control party. The DOOW, COW, Helm and planesmen receive valuable training in Dive Trainers which are located at major submarine training facilities such as Submarine School, New London, Connecticut and Trident Training Facility, Naval Submarine Base, Bangor, Washington. The ship's Navigator and piloting team can train on the Submarine Piloting and Navigation (SPAN) training system, also available at major submarine training facilities. However, the only training available for submarine officers to build or improve their shiphandling skills is on an actual submarine. For junior officers, this requires a great deal of close supervision by senior wardroom members. There is no margin for "training mistakes."

The submarine force needs to take advantage of the training possible using a virtual environment. In such virtual environments, junior submarine officers can receive valuable and realistic training experiences in which they safely learn from their mistakes.

C. OBJECTIVES

1. The Control Party

Like any other ship, the operation of a submarine, involves teams and teamwork. To gain the full benefits of team training in a virtual environment, several users must participate. These personnel must be able to exchange verbal orders associated with navigating the ship in a safe manner. Qualified watchstanders for each watch station assume the watch on a rotating basis in accordance with a posted watch bill. Specifically, the following watchstations need to be simulated.

a. Officer of the Deck (OOD)

The Officer of Deck (OOD) is in overall charge of the safe operation of the submarine. As such, he acts as a direct representative of the Commanding Officer (CO), handling the daily routine of operating the ship. The Officer of the Deck issues verbal orders to the Diving Officer of the Watch related to changing and maintaining the desired depth of the submarine. He issues orders to the Chief of the Watch related to various ventilation operations and watch section administration. Orders are issued to the Helm to control operation of the ship's rudder and speed. The Officer of the Deck receives all of his watch standing training "on the job," usually by standing watch as Junior Officer of the Deck (JOOD) under a more senior officer acting as Officer of the Deck. This training typically continues until the junior officer achieves an acceptable level of proficiency and is qualified by the ship's Commanding Officer.

b. Diving Officer of The Watch (DOOW)

The primary duty of the Diving Officer of the Watch (DOOW) is to reach and maintain ordered depth. The DOOW acknowledges and carries out orders from the OOD associated with ship's depth. The DOOW is also responsible for routine compensations to account for changes in the ship's buoyancy characteristics due to operations that affect ship's ballast. He typically supervises two planesmen in maintaining ship's depth: the Helmsman in the operation of the Engine Order Telegraph (EOT), rudder and forward (Fairwater or Bow) planes, and a second planesman in control of the stern planes. He formally supervises the Chief of the Watch for ballasting operations only.

c. Chief of the Watch (COW)

The Chief of the Watch (COW) acknowledges and carries out orders from both the DOOW and OOD. In addition the COW receives most of the routine, non-ship handling related communications and supervises the enlisted members of the watch section.

d. Helmsman

The Helm acknowledges and carries out orders from the OOD related to ship's speed and rudder control. He also acknowledges and carries out orders from the DOOW related to control of the Fairwater or bow planes. This person stands watch at the inboard ship control station.

e. Planesman

A second planesman acknowledges and carries out orders from the DOOW related to control of the stern planes and ships angle (pitch). This person stands watch at the outboard ship control station.

2. Hydrodynamics Model

The existing model does not prevent the user from driving the submarine into unrealistic situations, in particular plus or minus 90 degree pitch angles. Since the model uses an Euler Angle approach, when this situation occurs, the resulting mathematical singularity induces erratic model behavior. Also, the current model allows roll angles of up to 90 degrees. The model should "clip" the pitch and roll angles to reflect the behavior of an actual submarine. Since NPSNET is distributed openly, all of the source code is unclassified. Therefore changes made to hydrodynamic coefficients must remain unclassified.

D. THESIS OUTLINE

The preceding sections of this chapter outline the background and motivation for improving the submarine model in NPSNET. The remaining chapters are broken down as follows:

- Chapter II surveys related work including background on NPSNET, implementation of NPSNET ship and submarine simulators, and the Virtual Environment Submarine OOD Harbor Ship Handling Training System (VESUB) under development by Naval Air Warfare Center Training Services Division (NAWC-TSD), Orlando, FL.
- Chapter III presents an overview of the NPSNET Submarine Simulator.

- Chapter IV describes the design and implementation of a single remote control panel.
- Chapter V describes the mounting of human entities to the submarine to act as control party watchstanders.
- Chapter VI outlines conclusions reached as a result of this research as well as recommendations for future research in this area.
- Appendix A describes the Multi Gen OpenFlight model database hierarchy for the submarine visual model.
- Appendix B describes the video demonstration of the NPSNET Submarine Simulator.
- Appendix C describes how to obtain NPSNET source code, various documents related to NPSNET, and this thesis along with others related to the development of NPSNET.

II. RELATED WORK

A. INTRODUCTION

This chapter examines previous research which is pertinent to the NPSNET submarine simulator. A brief overview of NPSNET is provided which gives background on its development as well as how NPSNET entities interact with one another in a large scale virtual environment (LSVE).

B. NPSNET

NPSNET is a real-time, low-cost battlefield simulator that currently runs on commercial off-the-shelf (COTS) workstations from Silicon Graphics Incorporated (SGI) IRIX family of computers.Work on NPSNET was begun in 1990. Students and faculty in the Computer Science department were involved in the initial development and have made numerous improvements since this time. NPSNET_IV.10.3 is the current version of the evolving NPSNET simulation system. The Distributed Interactive System (DIS) Protocol version 2.0.4 is used for networked communications, and follows the object-oriented programing paradigm for defining and controlling remote and local DIS-based entities and munitions [Barham94].

DIS uses Protocol Data Units (PDUs) for sharing the events of a simulation amongst different host sites. There are twenty-seven standard defined PDUs for DIS which allow players to participate over local area networks (LANs) or multicast globally over the Internet. Only three of the PDU types are used by NPSNET: Entity State, Fire, and Detonation. There are no PDUs available in the DIS standard to describe the initial state of the simulation. Participants must agree beforehand what the initial exercise state will be. For example, the participants must agree on which terrain to use so that they all see the same geographical background such as ocean and land terrain.

Key information included in the Entity State PDU includes entity position and orientation (together referred to as posture), velocity and acceleration. All vehicle postures

are with respect to a global coordinate system common to all entities. A Fire PDU is sent by an entity when it fires or drops a weapon. A Detonation PDU is sent by a munition entity when it detonates or by an entity when it collides with a static or fixed object such as the ground terrain.

NPSNET-IV uses a Euclidean coordinate system (Figure 4) with axis x, y, z and Euler orientation angles for yaw (rotation about the z axis), pitch (rotation about the x axis) and roll (rotation about the y axis). Six spacial degrees of freedom are possible with combinations of translation and rotation in this coordinate system.

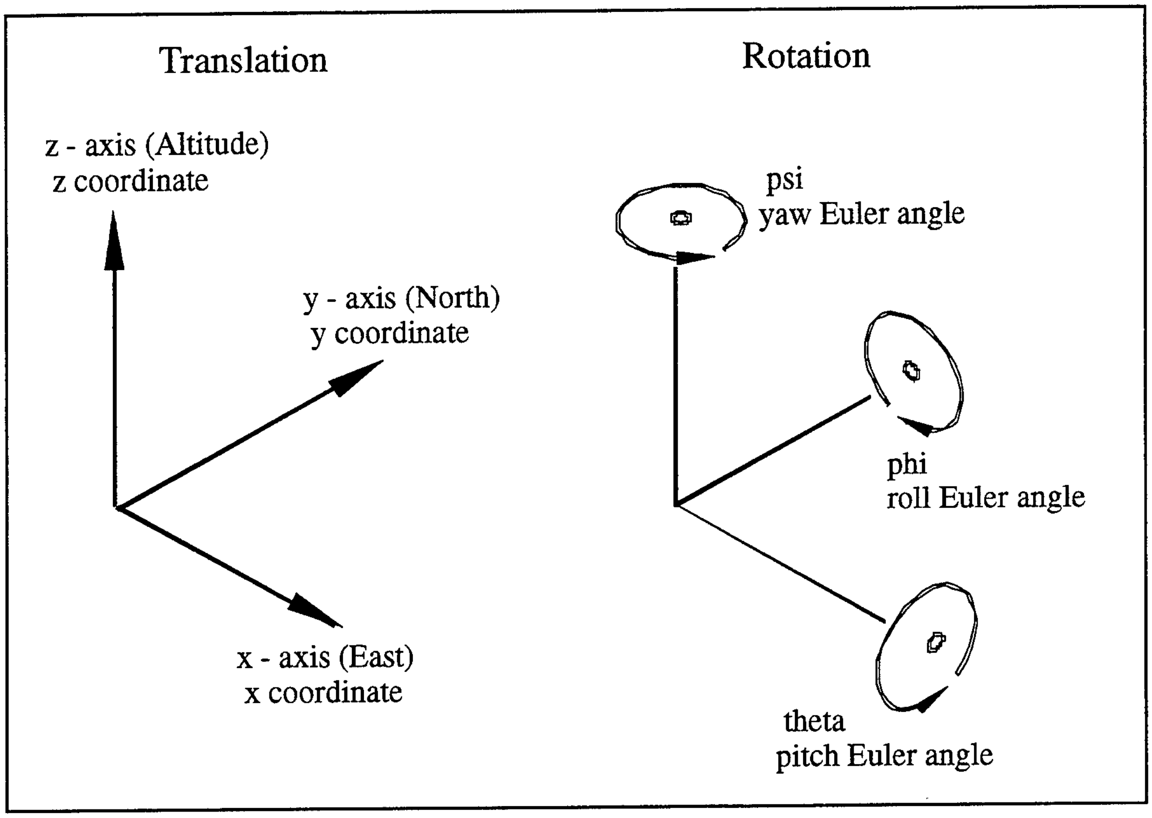

Figure 4: NPSNET Coordinate Axis Convention

To avoid network saturation, the PDU's are not send out continuously from each entity. Instead, each simulator sends out a PDU whenever its state changes significantly, such as when it starts or stops motion or makes a significant speed or course change. PDU's are also sent every five seconds (heartbeat PDU) if no significant changes have

occurred to inform other entities involved in the simulation that the entity is still alive. Between receipt of PDU's, each simulator dead reckons (DR's) the position of the other entities (ghost entities) in the simulation. The entity also dead reckons itself and when this estimated position deviates from its actual position by a set amount, a PDU is sent to update its position.

C. NPSNET SUBMARINE SIMULATOR

The Submarine Simulator in NPSNET use a real-time hydrodynamic model to calculate the submarine's movement through the virtual world. A rigorous general model for submerged vehicle hydrodynamics suitable for real-time simulation was created in 1994 [Brutzman 94]. This model was developed to support computer simulation and testing of the NPS Autonomous Underwater Vehicle (NPS AUV). This model utilizes standardized equations of motion, operating in real time (10 Hz or faster control loop). It was the first publicly available hydrodynamic model based on these standardized equations of motion. This model employs the object-oriented design paradigm, which allows programmers to implement and adjust for various types of submarines. A drawback of this model was that it assumed the submarine remained submerged at all times, with the buoyancy of the submarine remaining neutral [Bacon 95]. The model uses no less than one hundred separate coefficients which must be empirically determined for each type of submersible vehicle, hence extensive development is required to apply the model to a specific submarine types such as the Los Angeles (SSN- 688) or Trident (SSBN-726) classes.

Hydrodynamic coefficients which describe a general submarine form which is 14.292 feet long and weighs 1556 pounds were obtained from David Taylor Research Center, Ship Hydrodynamics Department [Bacon 95]. These coefficients were then scaled up to the 360 foot, 6,900 ton Los Angeles (688) class submarine. However, these coefficients were never actually implemented into NPSNET and therefore have not been tested. Instead, the current hydrodynamics model for the submarine simulator utilizes the coefficients that

were empirically determined for the NPS AUV [Brutzman 94]. At eight feet in length and weighing 435 pounds, the AUV is not a realistic model for the 688 class. The submarine model operates at very low speeds and is prone to erratic pitch and roll angles. It is possible to drive the submarine to pitch and roll angles of up to plus or minus 90 degrees. Since the hydrodynamic model uses Euler Angles for angular motion, when pitch angles of 90 degrees are reached mathematical singularities result and the vehicle motion is erratic and unpredictable.

The vehicle hydrodynamics model has been changed in the Submarine Simulator to account for changes in buoyancy as the ship broaches the surface. As the submarine begins to broach the surface, changes in the buoyancy are approximated utilizing the sine of the pitch angle [Bacon 95]. Actual changes in buoyancy require the calculation of a triple volumetric integral, which would be very expensive computationally and hinder the ability of the simulator to run in real time. For this reason the triple integral method was not implemented. The buoyancy model is accurate in the boundary conditions and provides a realistic simulation of buoyant pitch and roll at the surface.

The submarine simulator employs a multi-controller network protocol which allows multiple users at different workstations to control the same submarine entity via a control panel interface [Bacon 95]. This control panel interface includes three tabs: one each for the Officer of the Deck, Helm, and Weapons (Figures 5 - 7). The functionality of these panels is extremely limited however. Although command data is passed from the control panel to the submarine entity when the user selects the various control icons on the panel, the submarine entity does nothing with the data. For example, when the OOD selects the icon on his tab to raise periscopes the data is sent to the submarine entity but no functionality has been implemented in the submarine entity to actually raise the periscope and change the viewpoint of the user. Of the three tabs, the only command that is processed by the submarine entity is the ordered bell from the Helm tab.

An "ocean carpet" was implemented as part of the submarine simulator to simulate the motion of waves against the submarine hull. The carpet is positioned in the x and y axis and

rotates around the z axis corresponding with the heading of the local entity [Bacon 95]. Therefore, the ocean carpet follows the vehicle wherever it goes in the virtual undersea world while always remaining suspended one meter above the normal ocean polygons. The ocean carpet does not account for local swells or motion of the submarine through the water.

For user interfacethe submarine rudder is controlled by use of the keyboard (right/left arrow keys) or flight control stick (joystick). The up/down keys of the keyboard or the joystick of the flight control system control the deflection of the stern and forward (fairwater or bow) planes. Both sets of planes are positioned at the same angle of deflection, but in tandem at opposite directions. A rise angle on the forward planes corresponds to a counterclockwise rotation of the planes about the local x axis while a rise angle of the stern planes corresponds to a clockwise rotation of the planes about the local x axis. The user does not have the ability to control the forward and stern planes separately, making precise depth control difficult.

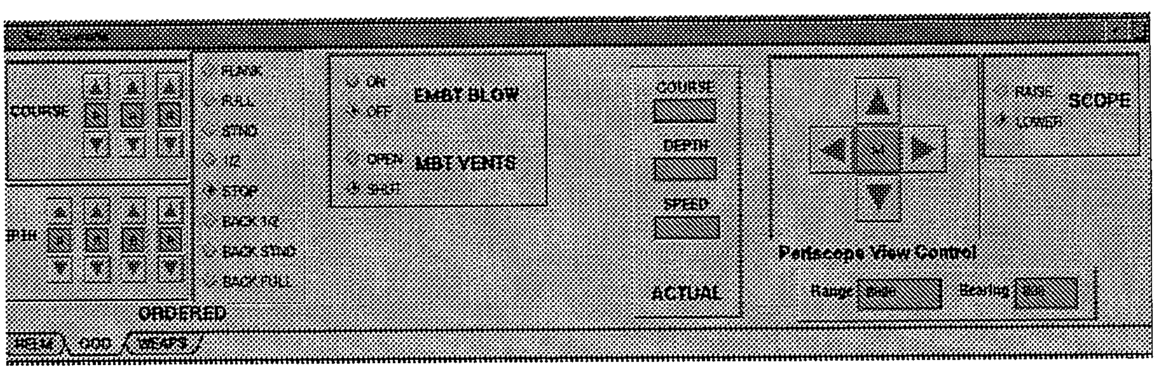

Figure 5: Helm Tab of Submarine Control Panel

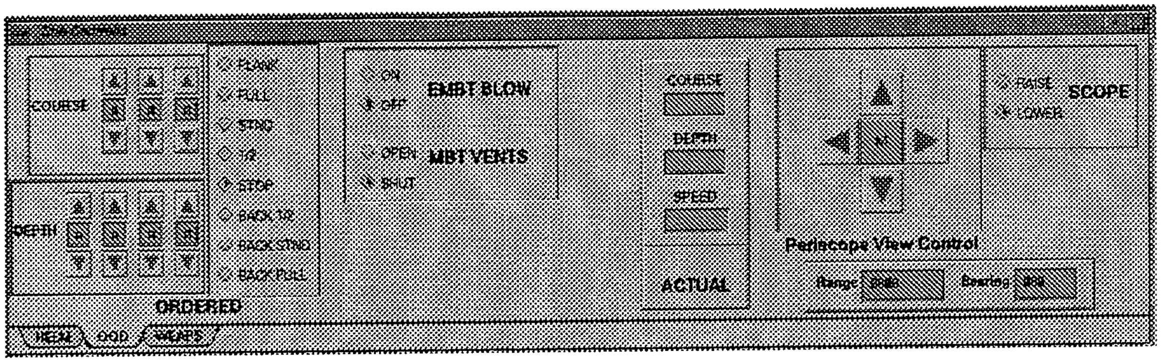

Figure 6: OOD Tab of Submarine Control Panel

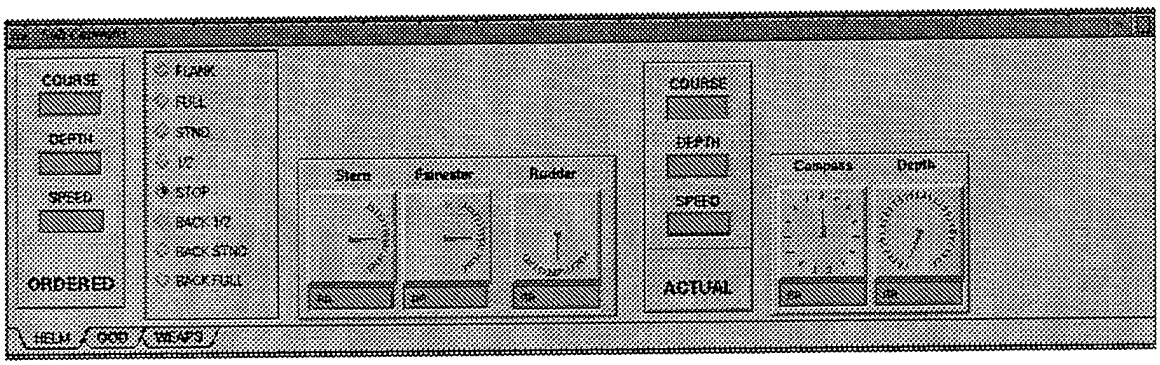

Figure 7: Weaps Tab of Submarine Control Panel

D. NPSNET SHIP SIMULATOR

The NPSNET Ship Simulator merged together the NPSNET Shiphandling Training Simulator (SHIPSIM) [Noble 95] and the NPSNET Damage Control Virtual Environment Trainer (DC VET) [Obyrne 95]. SHIPSIM was used to model the exterior of the ship while DC VET was used for interior modeling. This work successfully mounted human entities to ships such that they could interact with their surroundings. Specifically, human entities can open some doors, pick up a fire hose nozzle fight a fire, or even open and shut a valve to isolate a leak.

This simulator implemented voice recognition using SGI's Speech Manager [SGI9D 95]. This is a Dictation System, designed to recognize phrases and perform specified functions. The Speech Manager requires the user to train it on a vocabulary, either prior to

initial usage or when it does not properly recognize certain words in the vocabulary [Stewart 96]. The voice recognition is part of the ship control panel interface (Figure 8).

The model of the ship was developed from the Antares model, a commercial vessel model available from NAVSEA. The interior areas of the Antares model were modified to be representative of interior areas of an actual surface ship [Obyrne 95]. A large-scale virtual environment (LSVE) such as the interior of a ship requires efficient display of the environment to maintain a satisfactory frame rate. In the NPSNET Ship Simulator a Potentially Visible Set (PVS) algorithm is used to trim the database of non-visible polygons in each frame [Stewart 96]. In the case of the ship model, the database model is divided into spatial volumes or "cells." From within a given cell, there are a limited number of other cells that can potentially be seen from that cell. A good example of this is an office building where only adjacent halls and rooms are visible to one another. In the ship simulator, potentially visible cell list data are pre-loaded into the appropriate cell nodes of the model [Stewart 96]. Prior to drawing each frame, the viewpoint or driven vehicle position is checked against the bounding volume of the current cell. If the viewpoint is within the cell, no action is taken and the routine is exited. Otherwise, a list of all the cells in the model is searched until the viewpoint falls within the bounding volume of one of the cells. If no cell is found, the exterior of the ship is displayed.

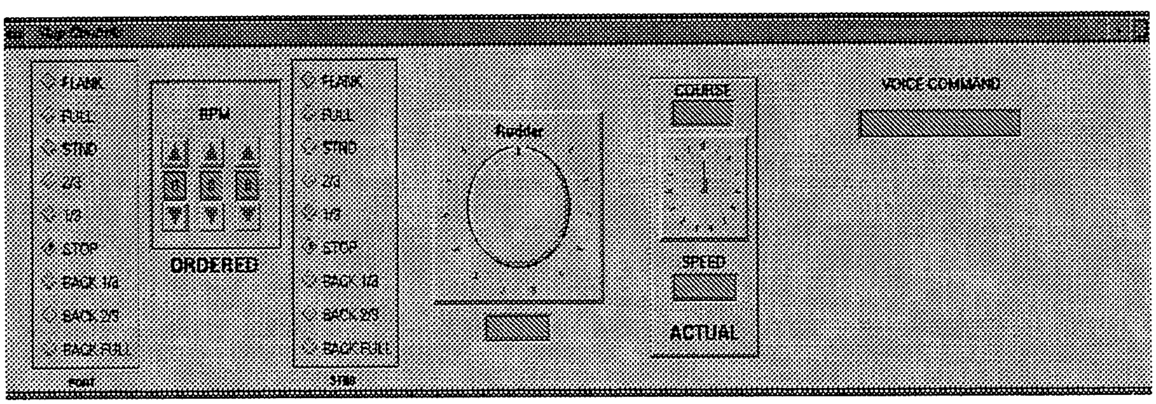

Figure 8: Ship Simulator Control Panel

Users can interact with the virtual ship using a variety of interfaces. A single control panel (Figure 8) is provided for inputting various shiphandling variables including ship's speed (bell orders) and rudder angles. Alternatively, a user can issue shiphandling orders via a microphone to the computer. Speech recognition software is used to "decipher" the verbal commands and effect the appropriate behavior of the ship's hydrodynamic model.

E. VESUB

The Virtual Environment Submarine OOD Harbor Ship Handling Training System (VESUB) is a system under development for training submarine OODs using Virtual Environment (VE) and speech recognition technology [NAWC 96]. The lead agency involved in the development of VESUB is the Naval Air Warfare Center-Training Systems Division (NAWC-TSD), Orlando, Florida. The system is being designed to provide valuable training to junior submarine officers for Officer of the Deck (OOD) qualification. Specifically, surfaced shiphandling training will be provided for harbors and channels. Delivery of this system to the fleet is expected in late 1998.

Limited steaming time is available to train these junior officer due to a shrinking submarine force and limited resources. Due to these limitations and the recent availability of Virtual Environment technology, the Navy has recognized the potential of a land-based simulator to train OOD's. In this system, the OOD trainee stands inside a rail box which restricts the lateral motion of the trainee to provide the feeling of actually being on the bridge of a submarine (Figure 9). The user is provided a 360 degree view of the virtual world through an HMD (Figure 10). A Polhemus Fastrak sensor is used to track the trainees head movement so that the appropriate scene can be rendered. The virtual world includes all the normal cues associated with harbor and channel navigation such as channel markers, navigation aids, surrounding landscapes and varying environmental conditions (Figure 11). The trainee can also view the bridge area and observe ship control indications from a simulated "bridge suitcase" (Figure 12). On an actual submarine, the bridge suitcase is a removable enclosure that houses indicators for ship's bell and course (with a removable

compass) and Interior Communication (IC) control switches. Attached to the bridge suitcase is a removable microphone which the OOD uses to communicate with various watchstations by manipulating the IC control switches to access appropriate communications circuits.

In the VESUB system, the OOD issues orders by voice into a microphone which resembles the type found on an actual submarine bridge. The voice orders are then processed by a speech manager which utilizes the HARK speech recognition system from Bolt, Beranak and Newman, Inc. (BBN) [Pioch 95]. The user can also use voice commands to change the current scene displayed on the HMD. For example, the command "binoc" causes a binocular view to be displayed which shows the surrounding scene at a magnification of seven times the normal view.

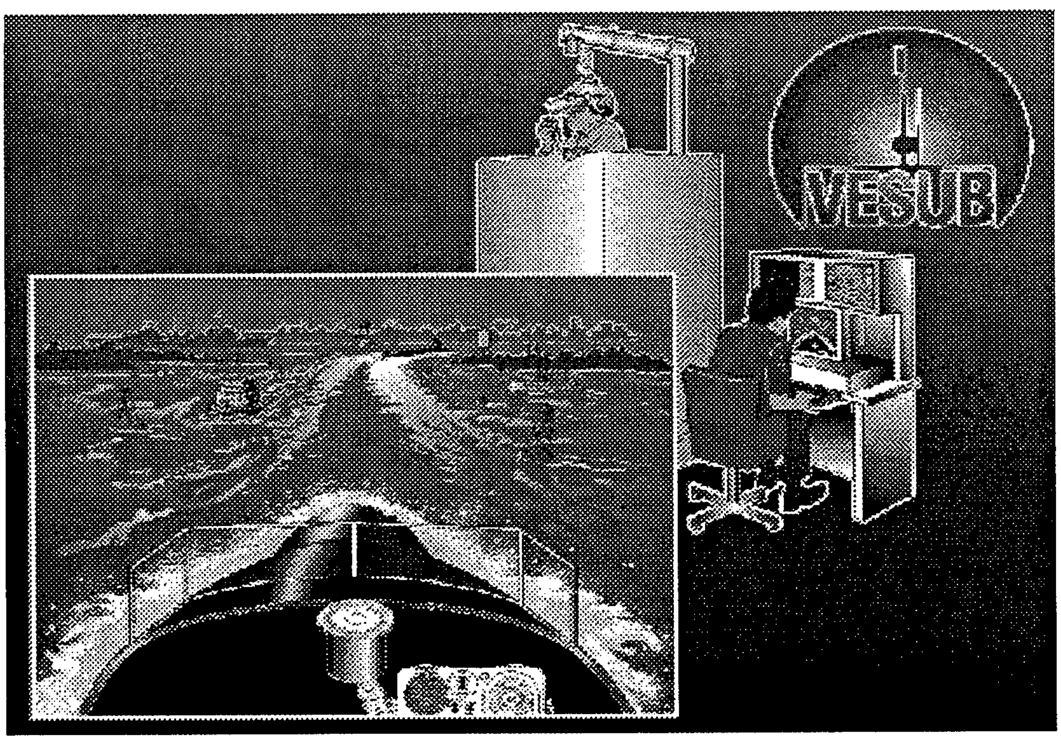

Figure 9: Artist's Representation of VESUB System [VESUB 97]

Figure 10: HMD Headset in VESUB [VESUB 97]

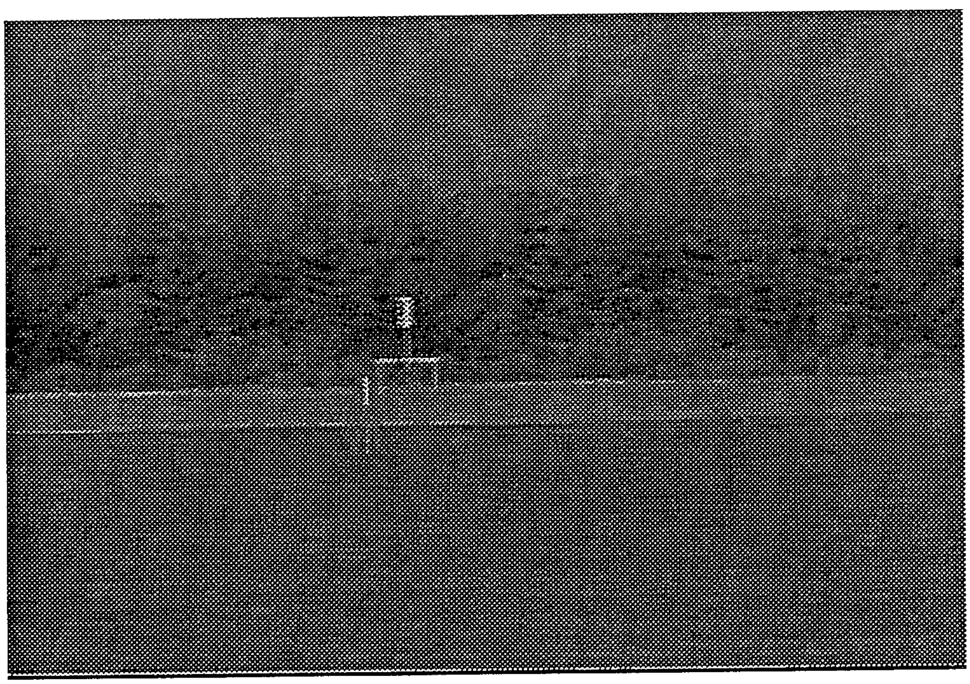

Figure 11: HMD Harbor View [VESUB 97]

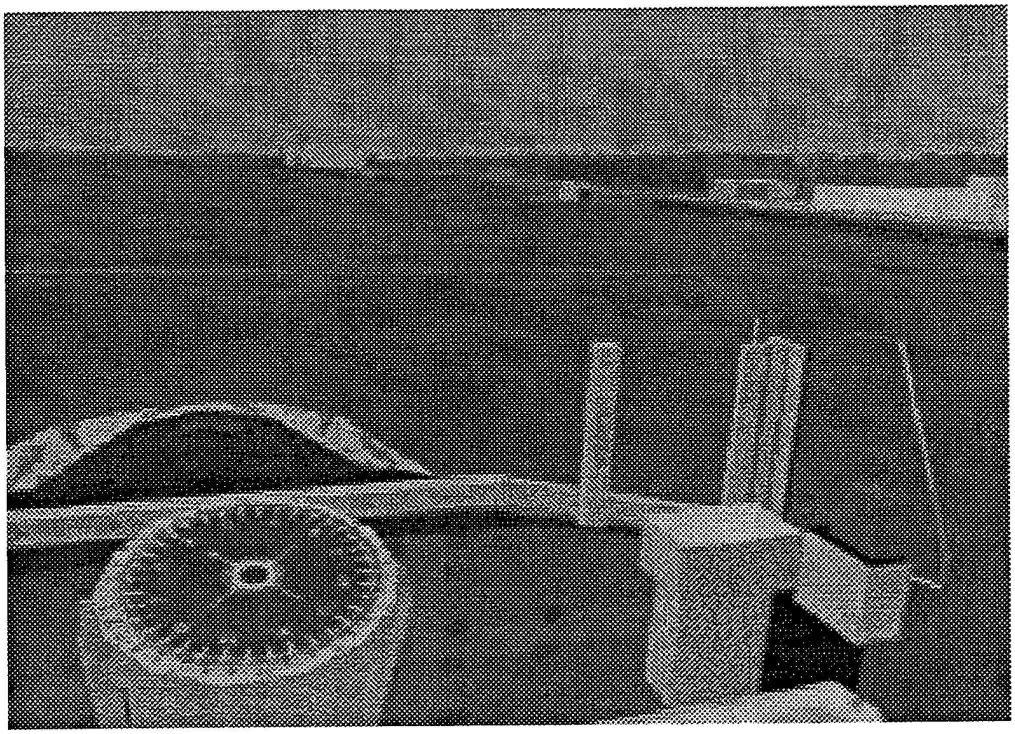

Figure 12: HMD Bridge View [VESUB 97]

The system is being built with two modes of operation: a stand-alone mode and an integrated mode. In the stand-alone mode, the student is able to converse with an automated helm, navigator, and other watchstanders through the speech recognition/synthesis system [MAWC 96]. Commands issued by the trainee result in appropriate changes to the visual scene and instrument readings (e.g. on the simulated bridge suitcase). The integrated mode physically links VESUB to the Submarine piloting and Navigation training System (SPAN) to provide training for the OOD in conjunction with the navigation team. The student then communicates with both the navigator and the instructor, who customarily plays the roll of the helm in SPAN exercises.

Harbors and channels are being modeled for approaches to the submarine bases at Kings Bay Georgia, Norfolk Virginia, and Bangor Washington. The initial version for Kings Bay Georgia is currently being tested at NAWC-TSD. The current system is capable of operating in the stand-alone mode only. The role of the helm has been automated but not the navigator. Complex ship handling orders such as "Helm, Bridge, left 20 degrees rudder,

steady 270" are not fully supported. In this case the system defaults to a hard left rudder, and proceeds to the ordered course.

A major drawback of the system is related to the HMD. The current technology used has a time lag between head/HMD movement and graphics scene updates. This results in tim- lag-induced simulator sickness which precludes the trainee from effectively using the system for more than 10 to 20 minutes at a time.

F. SUMMARY

NPSNET is a low-cost virtual environment which utilizes the DIS protocol for communications between entities in a simulation. DIS PDU's are utilized to communicate important information such as entity position, velocity, acceleration and orientation to the other entities involved in the simulation. To avoid network saturation, PDU's are not sent continuously. Between receipt of PDU's, each workstation Dead Recons the positions of the other entities involved in the simulation.

The NPSNET submarine simulator utilizes a hydrodynamics model that is physically based and runs in real-time. The interface employ a series of three control panel tabs which currently provide very little functionality. An ocean wave carpet model is used to simulate the motion of waves against the submarine's hull when it is on the surface.

The NPSNET ship simulator employs the Antares, a complete internal and external model of a ship. Human entities can mount this ship and manipulate various objects onboard such as a fire hose nozzle to combat a simulated fire casualty. Voice recognition is provided to allow the user to drive the ship via voice commands into a microphone.

The VESUB system is an effort to exploit the advantages of training in a virtual environment for submarine OOD's. Junior officers can receive valuable shiphandling training under surfaced conditions without getting underway on an actual submarine. Training on actual submarines is becoming more limited due to less steaming time available as the Navy continues to drawdown its forces.

III. SYSTEM OVERVIEW

A. INTRODUCTION

To improve the immersion qualities of the NPSNET submarine simulator a more realistic interface is needed. Human entities that are mounted to the submarine immerse the user into the environment as a human figure. Providing the human entities with the capability to control and maneuver the submarine provides a more realistic interface than pushing buttons on a remote panels which are physically removed from the virtual world. This chapter outlines the requirements necessary to build this interface.

B. DESIGN PHILOSOPHY

The NPSNET submarine simulator has been modified to allow human entities to mount and interact with the submersible entity. The submarine can be controlled by either a single user utilizing a single control panel which runs as a separate process or by a group of human entities, acting together as a team of watchstanders. The human entities are able to control the motion of the submersible vehicle. The submarine's speed, depth, and course are changed by interacting with simulated control icons at the individual watchstations.

The previous version of the NPSNET submarine simulator provided a realistic visual model of a LOS ANGELES (SSN-688) class submarine. This version suffered from the lack of a functional, realistic interface. The simulator employed an interface with separate control panel tabs for the OOD, Helm, and Weapons Officer that utilized a multi-controller protocol such that several persons could control the same submarine entity from separate workstations. This interface has very little functionality. Even though several control icons are available, callback functions are not implemented to affect the behavior of the submersible entity. This interface has been replaced by a team of mounted human entities who perform functions similar to those intended to be performed by these control panel tabs. Also, the submersible vehicle may be controlled by a single user utilizing a single control panel. This allows the user to train on his own.

The previous version of the NPSNET submarine simulator provides a rigorous hydrodynamic model such that the movement of the submarine through the water is physically based. This model employs no less than one hundred separate coefficients [Brutzman 95] representing drag, rudder and plane moments, etc. However, coefficients that are actually representative of the 360 foot, 6900 ton LOS ANGELES class submarine were needed. An appropriate set of coefficients has been previously developed from unclassified sources [Bacon 95] but never actually implemented in the submarine entity in NPSNET.

The LOS ANGELES coefficients were tested in the submarine entity utilizing the existing physically-based equations of motion. With these coefficients, the submarine entity moves in the forward direction (along body longitudinal axis) realistically as force is applied with the ship's propeller. This motion is realistic in that when a new bell is ordered (e.g. order ahead one third when ship is at all stop) there is an appropriate time lag as the ship comes to the expected speed. Intuitively, the coefficients related to drag along the body axis as well as force applied by the propeller seem to cause realistic hydrodynamic behavior of the submarine entity. Difficulty arises when the user applies rudder and/or planes (forward or stern) angles. With the submarine entity moving forward at 6 mph (5.28 knots), just one degree of rudder angle applied in either direction (left or right) causes the submarine entity to rotate and translate in all six degrees of freedom in an unrealistic and unpredictable manner. For example, the ship translates in the vertical direction such that the submarine is above the water surface. Similar affects occur when forward and stern plane angles are applied. Such behavior indicates sign or magnitude errors in the hydrodynamics coefficients. Further debugging is required.

The NPSNET Ship Simulator proved to be an excellent example of mounting human entities to a ship entity. Not only can human entities mount the ship, they can interact with several devices on the ship including doors, valves, and a firehose nozzle. It also employed several important features such as a potentially visible set (PVS) algorithm to improve

performance by limiting the number of polygons sent to the culling process during each cycle of the simulation loop.

C. NPSNET SUBMERSIBLE ENTITIES

In order to mount human entities to a submersible vehicle, the NPS submersible vehicle class required modification. Figure 13 shows the revised class hierarchy. An array of mounted entities was added as a data member to the Submersible Vehicle class as well as functions for mounting and dismounting from the submarine. This array of mounted entities (submariners) is updated each time the submersible vehicle is dead reckoned so that they maintain correct position relative to the submarine. Likewise, when the submarine entity receives an Entity State Update PDU, the human entities mounted to the submarine are updated. A new class was developed, Submersible Walk Vehicle that publicly inherits from the Submersible Vehicle class. This new class was added to provide the ability for mounted human entities, who can walk around the submarine entity, to manipulate various objects such as periscopes, rudder, and planes from a particular watchstation. For example, the helmsman can manipulate the rudder and Engine Order Telegraph (EOT) from his watchstation.

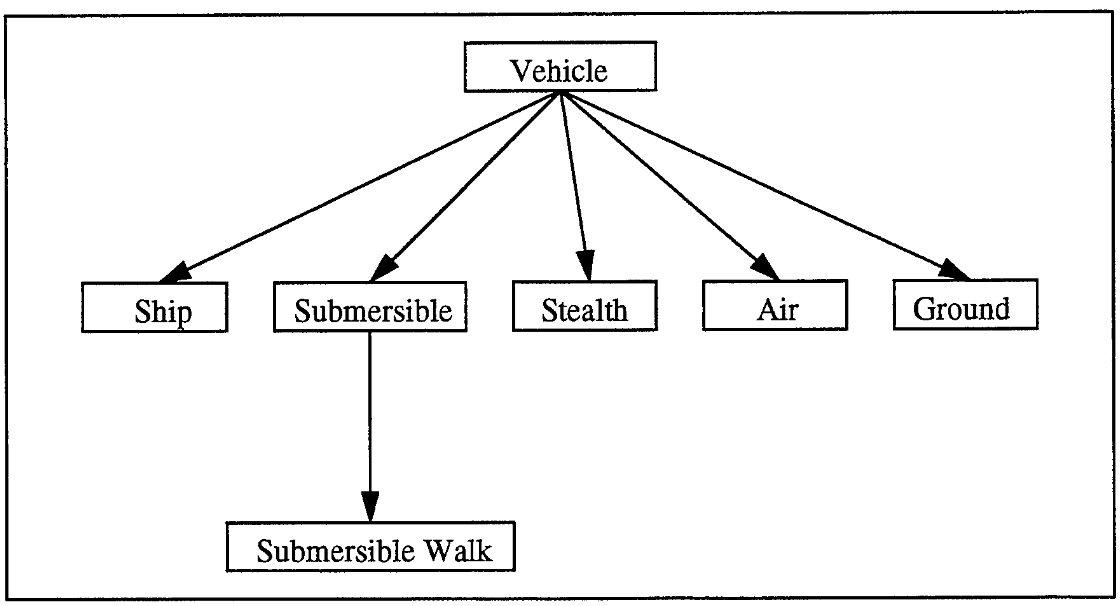

Figure 13: NPSNET Class Hierarchy for Vehicles

The submarine simulator utilizes the same object-oriented, physically based hydrodynamic model as the NPS AUV. Figure 14 outlines the class hierarchy of the NPS AUV model.The UUVBody class object reads coefficients from a separate file for the hydrodynamic equations of motion. A set of coefficients which represent a LOS ANGELES class submarine was developed for use in the UUVBody class; however, due to erratic model behavior this set of coefficients has been replaced with the NPS AUV coefficients. The UUVBody class implements all the necessary functionality to calculate positions, velocities, and accelerations in six degrees of freedom utilizing standardized equations of motion.

The submersible vehicle encapsulates, or includes as a data member, a UUVBody class object. The information provided by the UUVBody class is utilized to update data members of the submersible vehicle class including vehicle position and orientation. This information is then used to place the vehicle in the correct position, and at the correct orientation in the virtual world.

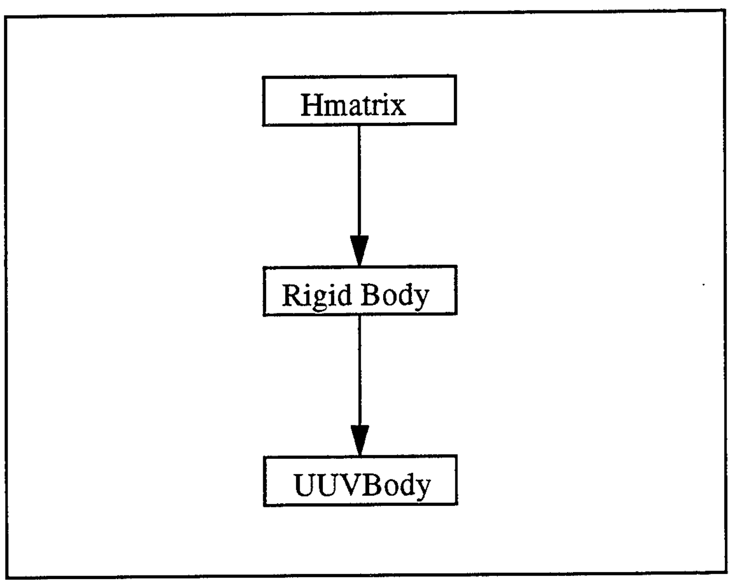

Figure 14: NPS AUV Hydrodynamics Model Class Hierarchy

D. MULTIGEN OPEN FLIGHT MODEL

MultiGen is a three dimensional (3D) modeling tool utilized for building models of complex objects such as ships, aircraft, and submarines. This system stores objects of the drawing in a hierarchical database which the user specifies. MultiGen supports a vast array of colors, textures, and material types.

MultiGen has the ability to build degree of freedom (DOF) nodes, which are used to move certain objects in the model. The user specifies during the model building process translations and rotations that a particular DOF node may undergo including limits on translation distances and rotation angles.

The existing MultiGen model was modified to add DOF nodes for the port and starboard periscopes, rudder, forward (fairwater) planes and stern planes. Also, a control room was added to the interior area of the submarine at the approximate location of an actual control room. Several objects and DOF nodes were then added to the control room for later manipulation by mounted human entities.

In group nodes, the user can store two data values (special1 and special2) which are useful for capturing certain nodes and data during the database loading process. The model database can then be stored in the OpenFlight (.flt) format, which can then be loaded by Performer, the principal API supporting NPSNET. Performer 2.0 includes the OpenFlight loader which allows the user to specify a callback function that is passed into the loading process. The user specifies in this callback function what, if any, action is to be taken as the loader is processing a node (beads in OpenFlight) of a particular type. This is where the special1 and special2 fields of MultiGen group nodes can be useful.

In the revised submarine simulator, when the loader sees the degree of freedom node for a periscope, which is identified by a code in the special1 field, it sets a pointer to the node so it can be used later. It also captures data, such as the current position of the periscope. Similar action takes place for the rudder, forward (fairwater) planes, and stern planes.

Besides DOF nodes, the callback function implemented for the revised submarine simulator also checks for Potentially Visible Set (PVS) nodes which are again identified by a special code in the special1 field. These nodes are used to build a list of PVS cells and which, if any, other cells are visible from that cell. The complete structure of the submarine model database is provided in Appendix B.

E. REMOTE CONTROL PANEL

A single user mode is provided with the NPSNET submariner simulator. This allows the user to train in an individual mode without a complete control party watch team. The interface utilized for this purpose is a single control panel, which runs as a separate process at a separate workstation. This control panel includes all the necessary functionality to change ship's speed, depth, and course. The user may also raise and lower the periscopes or change the bearing and attitude of the periscope view point.

F. NPSNET HUMAN ENTITIES

The submarine simulator multi-user interface has been updated to improve the immersive qualities of the simulation. Multiple human entities can mount the submersible vehicle to act as control party watchstanders representing the OOD, DOOW, COW, Helm, and Dive Planesman. These human entities can control the motion of the submersible entity through the virtual world by interacting with various manipulable objects at their watchstations.

The NPSNET ship simulator provided a good working model for mounting human entities to an entity that actually moves, but not in all six degrees of freedom. The human entities mounted to the ship move (walk) along a horizontal xy plane representing the various decks of the Antares ship model. The human entities then "snap to" the correct vertical (z) position by performing an intersection test against fixed and moving objects such as the ship. For example, if a the human entity encounters a ladder, an intersection test in the vertical direction indicates that the entities should snap to the height of the first step of the ladder and so on as the ladder is climbed. The human entity also rotates about the

vertical z axis (yaw) as the ship changes course so that the human's heading changes by the same amount and direction as the ship's heading.

Several changes were needed to mount the human entity to the submersible vehicle. As the submarine changes depth, the pitch angle of the ship changes as the user applies planes angles. With the ship approach, when this occurs the human entity stays at the same vertical position, thus sinking into or raising above the deck to which he is mounted. Thus, the vertical position of the human entity must take into account the pitch angle of the ship since he is mounted to the control room deck, which is forward of the submersible vehicle center. Likewise, similar changes are needed to account for roll which occurs when rudder angles are placed on the submarine.

G. HIGH-RESOLUTION NETWORK

In NPSNET, the DIS standard is used to perform the necessary network activities. The principal DIS packet used in a simulation is the Entity State PDU which is used to inform the players in the simulation of the state of all players in the simulation. Nevertheless there are additional pieces of information that need to be communicated in order to mount human entities to submersible vehicles that are not included in any of the DIS packets. NPSNET provides a different network, the High Resolution Network (HIRESNET) to satisfy this need [Stewart 96]. The HIRESNET sends Data PDU packets between entities on the network. Only those workstations listening the HIRESNET will display and process the information contained in the Data PDU.

The submersible vehicle entity utilizes the HIRESNET to communicate information such as raising and lower periscopes, and moving control surfaces. The human entities use this network to communicate mounting information. In order for the submarine simulator to become fully DIS compliant, the DIS standard needs to be expanded to provide the same functionality as the HIRESNET. Further information is provided in Section V.D.

H. SUMMARY

Improving the NPSNET submarine simulator interface is essential to make it a viable training tool. By mounting human entities to the submersible entity who can control and manipulate the submersible vehicle in the virtual world, the user is immersed in the simulation as a moveable human figure.

Improvements in the submarine Open Flight model were necessary to allow various visual objects to be manipulated in the simulation. Also, it is necessary to capture various nodes in the database hierarchy during the loading process so that they may later be manipulated or utilized to determine the appropriate PVS cell.

The DIS networking standard does not include all the data communication necessary to implement this improved interface. The ability to communicate essential data for mounted human entities and manipulated objects on the submersible vehicle entity is provided by the HIRESNET.

IV. SINGLE USER INTERFACE

A. INTRODUCTION

NPSNET supports a variety of interfaces including a flight control system (FCS) and the keyboard. Utilizing the keyboard, a variety of inputs are possible to control the environmental conditions of the simulation and manipulate vehicle entities [UG-IV.9 96]. However, there are no keyboard commands available to manipulate the submarine's forward and stern planes separately. Pressing the up or down arrow key causes both the forward and stern planes angles to change in the hydrodynamics model. For example, the up arrow causes a rise angle on both planes which corresponds to a counterclockwise rotation of the forward planes and a clockwise rotation of the stern planes. More keyboard commands are needed to allow for separate control of all the submarine's control surfaces. The user may also utilize the flight control system to control the submarine. Although the user has a separate control for the throttle, the use of a joystick for controlling both ship's course and depth is unnatural. Forward and aft motion on the flight control joystick causes the forward and stern planes to deflect in opposite directions, by the same amount. No changes have been made to the existing FCS interface.

In order to provide the necessary controls for the submarine simulator in single user mode, a graphical user interface (GUI) control panel was developed. The control panel is a more natural interface in that it provides control icons (e.g. dials and buttons) that are more descriptive and provide finer control.

B. KEYBOARD

A member function was added to the submersible vehicle class to define keyboard controls specific to the submarine. Keyboard commands were added in this function to allow for the separate control of the forward and stern planes. No commands have been added, however, for the control of the submarine's periscopes. Since the principal single

user interface is the control panel, control of the periscopes is only provided in this interface. Appendix A provides a listing of the submarine-specific keyboard commands.

C. CONTROL PANEL

To provide a single user interface that is more intuitive than the keyboard or flight control system, a control panel was developed (see Figure 15). The control panel was created using Developer Magic's RapidApp application builder [SGIC 94]. This GUI development environment contains many different types of widgets such as dial knobs, push buttons, input text boxes, and radio buttons. With these widgets, a control panel was built which resembles something that might actually be seen on a submarine. RapidApp was chosen for its ease of use.

A radio button is a series of toggle buttons arranged in a column. The user is allowed to select one (and only one) of the toggle buttons at a time. The remaining toggle buttons are turned off. The selected toggle button then initiates a callback function which performs the actions implemented for that toggle button. Radio buttons are provided for the ordered bell, for raising and lowering the periscopes, and for setting the current viewpoint. Selecting a bell toggle button on the ordered bell radio button changes the propulsion order. Selecting the raise toggle button on a periscope radio button causes the periscope to raise. Conversely, selecting the lower toggle button on a periscope radio button causes the periscope to lower. Selecting the PORT SCOPE toggle button on the current viewpoint radio button changes the current viewpoint to the port periscope eyepiece if that scope is currently raised. Figure 16 depicts a view from the port periscope with the submarine at periscope depth. Selecting the STBD SCOPE toggle button changes the current viewpoint to the starboard periscope if that scope is currently raised. Selecting the CONTROL ROOM toggle button changes the viewpoint to the control room. Thus, when one or more periscopes are raised, the user can choose whether to look out a periscope or maintain a viewpoint in the control room.

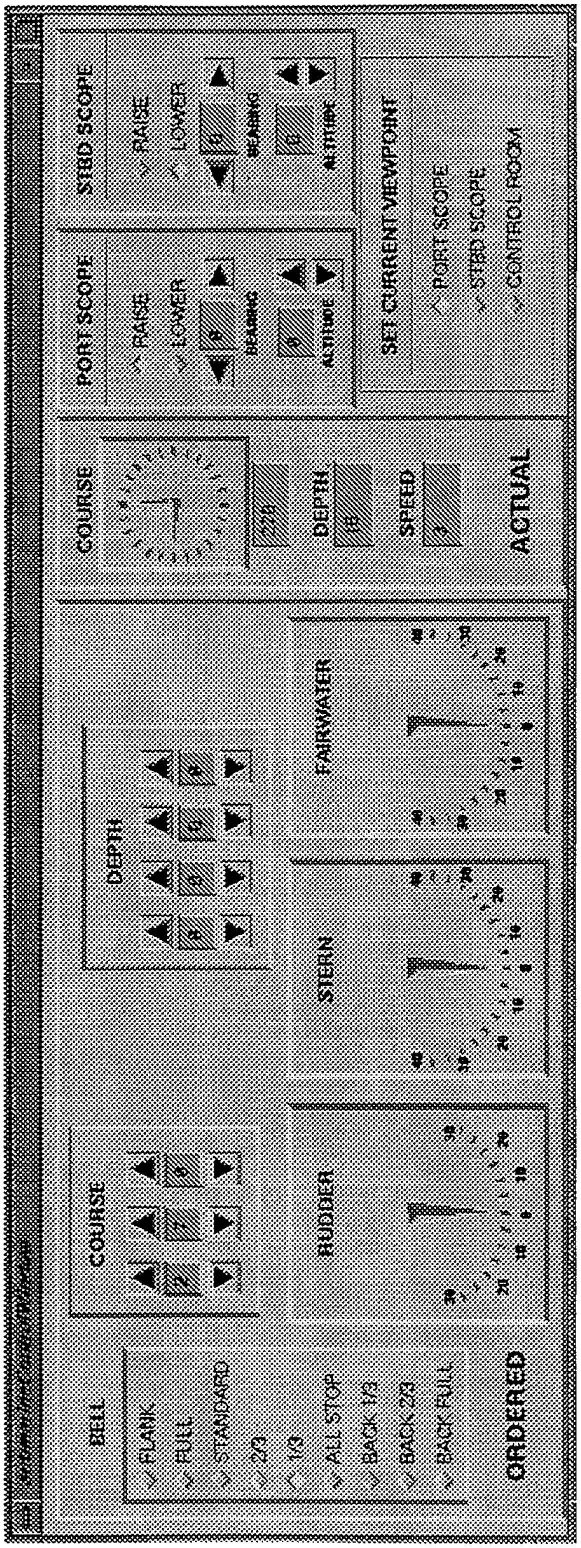

Figure 15: Submarine Control Panel

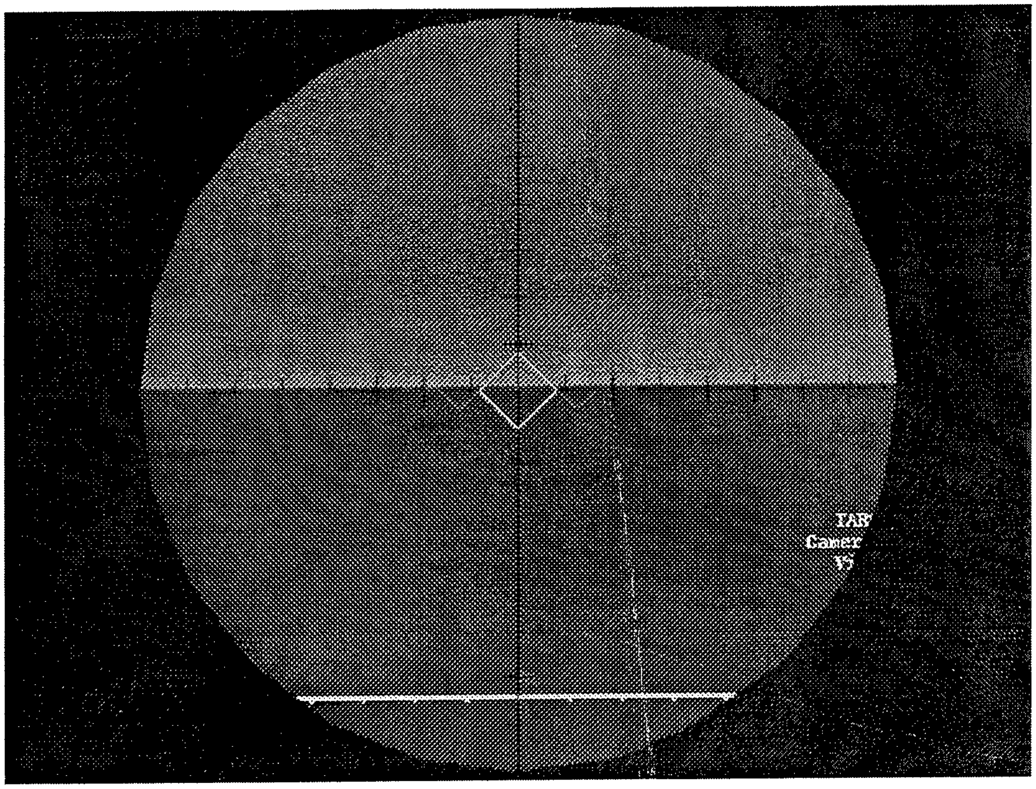

Figure 16: View from Port Periscope at Periscope Depth

A dial knob displays a range of angles that the user may select for the specified item. The user may change the angle by either selecting the red hash marks or by dragging the dial needle to the desired angle. When a new angle is selected, a callback function performs the implemented actions for that dial. Angle dial knobs are provided for the rudder, fairwater planes, and stern planes. Selecting an angle on one of these knobs changes the angle of deflection of the corresponding control surface. For example, selecting the red hash mark or dragging the needle on the stern planes dial knob to the hash mark just above the ten (to right of zero) causes ten degrees of rise on the stern planes.

Text fields are provided to display the actual speed, depth and course. In addition a dial knob is provided to display the actual course. Buttons are provided for inputting an ordered course or depth, but functionality has not yet been implemented for these inputs.

In order to communicate with NPSNET to control a submersible vehicle, the network protocol established previously for the submarine simulator is utilized (multi-controller protocol). In this protocol, the control panel communicates with NPSNET via a multicast port using a special PDU packet known an Information PDU (IDU). Conversely, NPSNET communicates with the control panel via an IDU. The IDU packet includes header information (see Figure 17) as well as a defined data structure. The data structure utilized by NPSNET to communicate with the submarine control panel is the NPSNETToSubIDU (see Figure 18) which includes actual parameters for the submarine entity such as actual course, depth, speed. The data structure utilized by the control panel to communicate

```
#define NPSNET_To_SUB_Type        (IDUType)103
#define SUB_Ood_To_NPSNET_Type  (IDUType)105

typedef struct {
   unsigned char    version;
   IDUType          type;
   unsigned short   length;
} IDUHeader;
```

Figure 17: IDU Packet Header

```
typedef struct {
   IDUHeader    header;
   // The following fields are used to communicate data from NPSNET
   //    to SUBCONTROL
   float sub_depth;    // actual, 0000
   float sub_course;   // actual, 000
   float sub_speed;    // actual, 00

   u_long space_holder;

} NPSNETToSubIDU;
```

Figure 18: IDU Packet Structure from NPSNET

with NPSNET is the SubOodToNPSNETIDU (see Figure 19). This IDU communicates orders such as ordered bell, course, forward and stern planes angles, rudder angles and periscope manipulations.

```
typedef struct {
    IDUHeader header;
    float ood_ordered_depth;        // ordered
    float ood_ordered_course;
    int ood_ordered_bell;

    float ordered_rudder;
    float ordered_fwdplanes;
    float ordered_sternplanes;

    int ood_stbd_scope_raiselower;
    int ood_port_scope_raiselower;

    float ood_port_scope_alt;
    float ood_stbd_scope_alt;
    float ood_port_scope_brng;
    float ood_stbd_scope_brng;

    u_long space_holder;

} SubOodToNPSNETIDU
```

Figure 19: IDU Packet Structure from Control Panel

D. SUMMARY

The principal single user interface for the NPSNET submarine simulator is the submarine control panel. This panel provides the user a realistic-looking control panel from which the submarine can be driven. In addition to changing the ship's course, speed, and depth, the user can also manipulate the port and starboard periscopes. Feedback is provided from the submarine entity to the control panel so that the user is aware of actual shiphandling parameters such as speed, course, and depth. This panel does not, however, provide for the manipulation of environmental variables such as time of day, lighting and fog density. These parameters may only be manipulated with the keyboard.

Secondary interfaces for the submarine simulator include the keyboard and the flight control system. With the keyboard, the user has separate control of both the forward and stern planes.

V. MOUNTING OF HUMAN ENTITIES

A. INTRODUCTION

Up until the development of the NPSNET ship simulator, all entities were independent of one another [Stewart 96]. Human entities moved through the virtual world, not attached to or associated with any other entities. These human entities walked or ran over terrain such as the ground and water. They were unable to walk onboard a ship, let alone manipulate various objects onboard the vehicle.

The ship simulator provided the important capability for human entities to mount a ship by walking aboard via a ramp, which extended down to the ground next to the mooring position of the ship. An intersection test is performed and when an intersection is detected between the human and ramp bounding volumes, the human entity mounts the ship. Once mounted, the human entity moves through the virtual world with the ship such that if the ship changes speed, for example, the relative position of the human entity with respect to the ship remains constant as long as the human entity is not moving. The human entity's global position is determined by adding a relative position vector to the ship's position vector. If the human entity is moving, its relative position to the ship is updated so that the human entity maintains correct position with respect to the ship. Since the ship is assumed not to change its pitch and roll angles, no capability is provided to ensure that a human entity maintains correct relative position onboard the ship as pitch and roll angles are varied. Only the azimuth, or heading of the ship was considered. This is a reasonable (although simplistic) assumption for a surface ship but changes are needed for a submersible vehicle which is subject to significant pitch and roll angles.

NPSNET is a networked virtual environment in which each workstation simulation maintains a "ghost" entity for every other entity participating in the simulation. These ghost entities are updated periodically when DIS Entity State PDUs are received. To limit network traffic, these PDUs are only sent when an entity changes acceleration, velocity or orientation significantly. Between receipt of Entity State PDUs, each workstation dead

reckons (DRs) the position of the other entities in the simulation. Effective dead reckoning algorithms ensure that the behavior of ghost entities is realistic, and that DR position does not differ drastically from actual positions. The DIS standard does not, however, provide the capability for human entities to inform their associated ghost entities to mount a vehicle such as a ship. In order for the ghost human entities to mount and maintain correct position onboard the ship, they need to be informed that they are mounted to a ship. This is accomplished by using the HIRESNET. Only those workstations listening for HIRESNET Data PDUs actually process them.

Once mounted onboard a submarine, the human entity immerses the user into the environment as a human figure. To control and maneuver the submarine through the virtual world, the human entities need to interact with various objects onboard the submarine in some manner. The submarine Open Flight visual model was first modified to include a rudimentary representation of a control room. Several manipulable objects were then added to the control room. As a result, the human entities can manipulate various objects onboard the submarine utilizing a simple "picking" mechanism. The user is able to position a set of cross hairs on an object utilizing either the keyboard or Flight Control System (FCS). When the cross hairs are centered on an object, and the user "picks" the item utilizing either the mouse or FCS, a pre specified action occurs. For example, the human entity can pick a periscope operating handle, which then causes the periscope to be raised. In order to simulate the actions of a complete watch team, several of these human entities are mounted to the same submarine, each assuming a particular watchstation. Specifically, the submarine's control party is simulated using human entities to represent the Officer of the Deck, Diving Officer of the Watch, Chief of the Watch, Helm and Dive Planesman.

B. CHANGING THE MULTIGEN OPEN FLIGHT MODEL

The existing submarine Open Flight visual model provided an exterior representation of a LOS ANGELES class submarine. Exterior surfaces such as the hull, sail, and planes surfaces were included. In order to provide the capability for human entities to mount the

submarine and manipulate various objects onboard, several modifications to the previous model were required.

A rudimentary control room was added to the model at the approximate location of an actual control room under the submarine's sail (see Figure 20). It consists of forward and aft bulkheads and a deck positioned such that the control room occupies approximately one third of the inner hull space vertically. In the horizontal direction, the control room occupies eight meters of the hull's inner space. Several objects were then added to the control room.

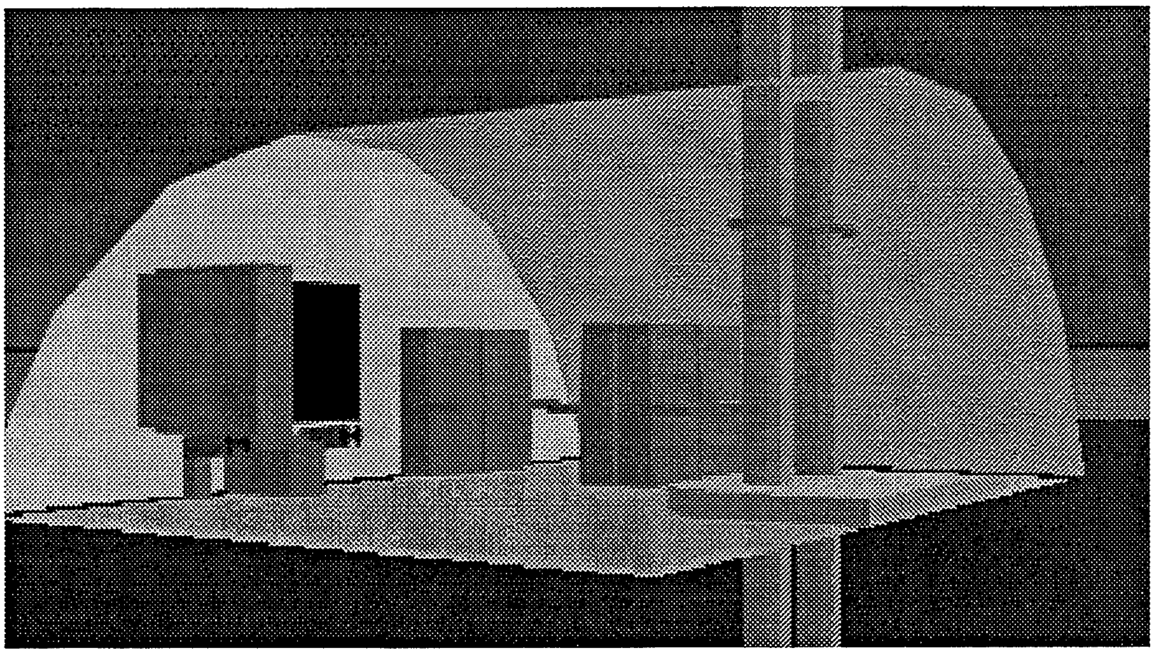

Figure 20: Submarine Control Room

1. Conning Station

The Conning Station is where the Officer of the Deck stands his watch. The Conning Station is on a raised platform at the aft (rear) end of the control room (see Figure 21). Included at the conning station are the periscope housings. On the upper part of each periscope housing is a hand wheel for operating the periscope. Picking a handwheel causes the associated periscope to either raise or lower depending on its current position. For example, if the periscope is currently lowered, picking the handwheel will cause it to rise.

When a periscope moves up or down, the housing moves with it. Figure 22 show the port periscope housing with the port periscope in the raised position.

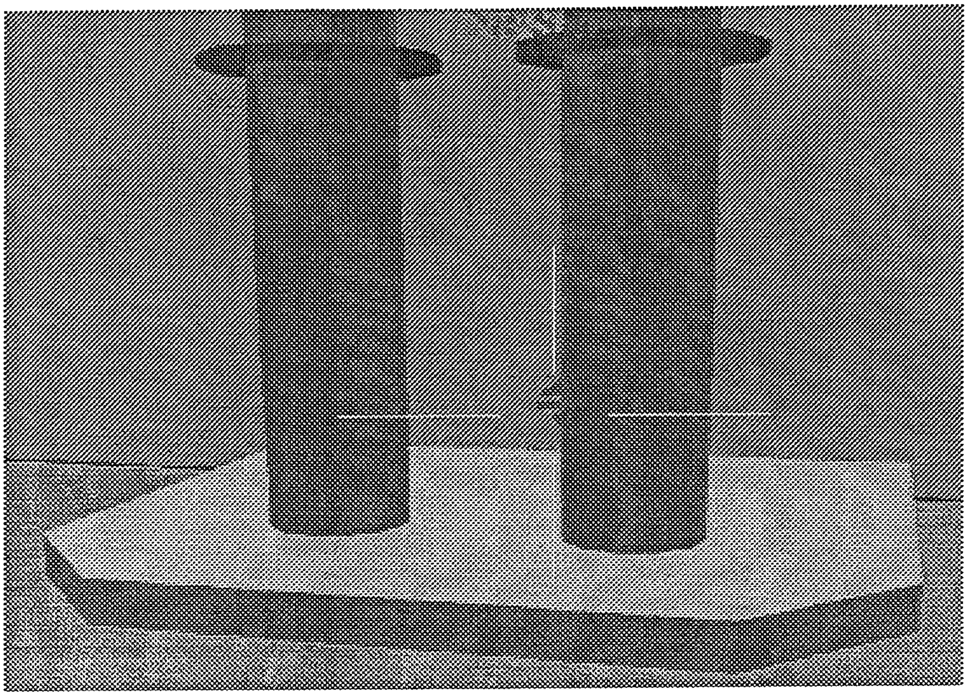

Figure 21: Conning Station of Submarine Control Room

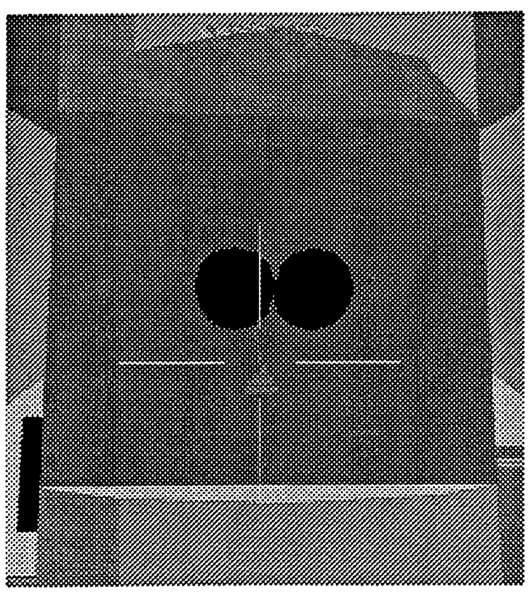

Figure 22: Port Periscope Housing

2. Helm Station

The Helm Station is where the helmsman/planesman stands watch. Provided at the helm station are an Engine Order Telegraph (EOT) for relaying propulsion orders to maneuvering, and a control stick for operating both the rudder and the forward planes (see Figure 23). Picking the steering wheel on the stick with the left mouse button causes the steering wheel to rotate counterclockwise, and the rudder to rotate clockwise (left rudder). The updated rudder angle is sent to the hydrodynamics model for calculating changes in the ship's position and orientation. Conversely, picking the steering wheel with the right mouse button causes the wheel to rotate clockwise and the rudder to rotate counterclockwise (right rudder). Picking either the vertical or horizontal links of the stick with the left mouse button causes the vertical link to rotate in a forward direction, the horizontal link to translate forward and downward, and the forward planes to move in the dive direction. Conversely, picking either link with the right mouse button causes the vertical link to rotate in the aft direction, the horizontal link to translate back and down, and moves the forward planes in the rise direction. Again, the changes in the forward planes angle are sent to the hydrodynamics model.

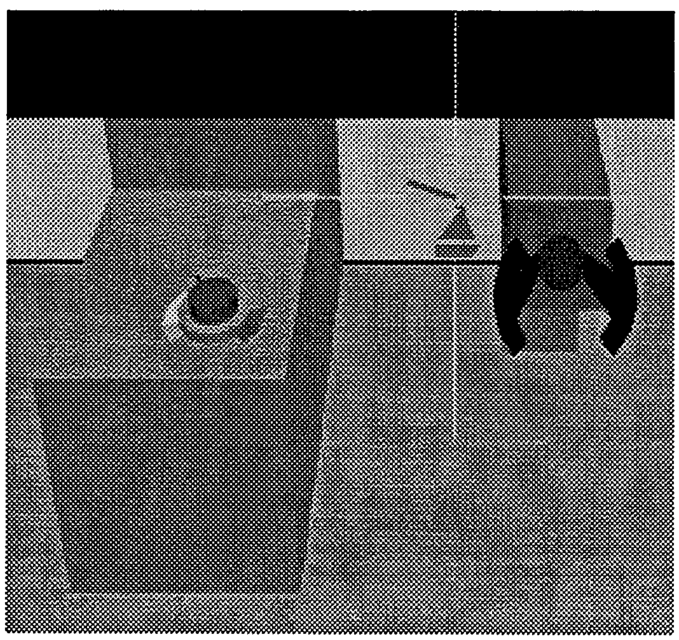

Figure 23: Helm Station of Submarine Control Room

3. Dive Planesman Station

The Dive Planesman station is directly to the left (outboard) of the Helm station (Figure 24). At this station is a control stick similar to the one at the helm station. Picking the vertical or horizontal links of the control stick causes the control stick and stern planes to move. However, the steering wheel on this control stick is non-functional. On an actual submarine, watchstanders can actually shift control of the rudder to the outboard station. Likewise, they can select which set of control planes is manipulated by each control station. For simplicity these features have been left out of the submarine simulator.

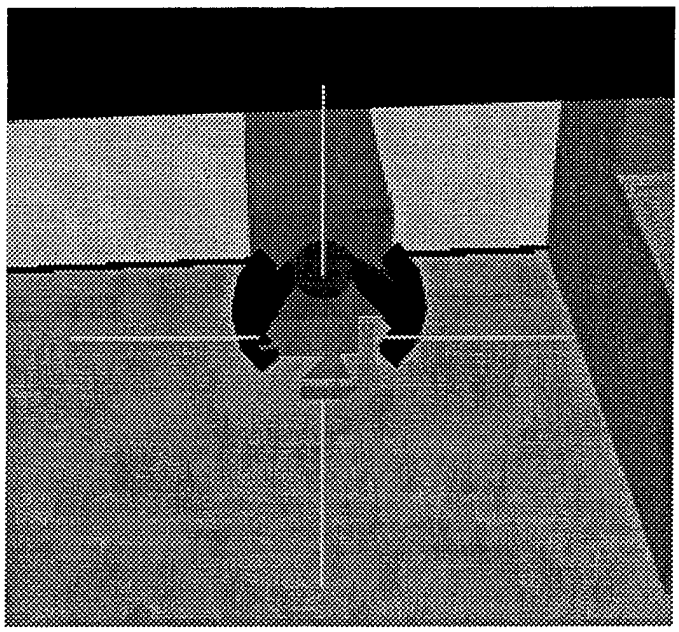

Figure 24: Dive Planesman Station of Submarine Control Room

4. Diving Officer of the Watch station

The Diving Officer of the Watch (DOOW) station is located directly behind the Helm and Dive Planesman watchstations. The large black panel directly forward of the helm and Dive Planesman watchstation is the Diving Officer Indicator Panel. On an actual submarine, various indications such as actual plane and rudder angles, depth, and speed would be available here. The DOOW issues orders to the Helm/Planesman and Dive Planesman to reach and maintain ordered depth. Currently, there is no voice recognition

capability in the submarine simulator for orders and acknowledgments to be processed and passed to the appropriate watchstation. This may be done with headsets and two-way radios, or by voice communication in an open room.

5. Chief of the Watch Station

The Chief of the Watch (COW) station consists of a vertical panel and a smaller attached horizontal panel (see Figure 25). The COW control various switches to add and remove ballast from the ship and perform various ventilation operations. On an actual submarine, the vertical black panel would include all of these control switches. The horizontal panel area serves as a desk space where the COW keeps track of various operations onboard the ship that affect ballast as well as a myriad of watch section administrative details. Currently there are no manipulable objects at this station.

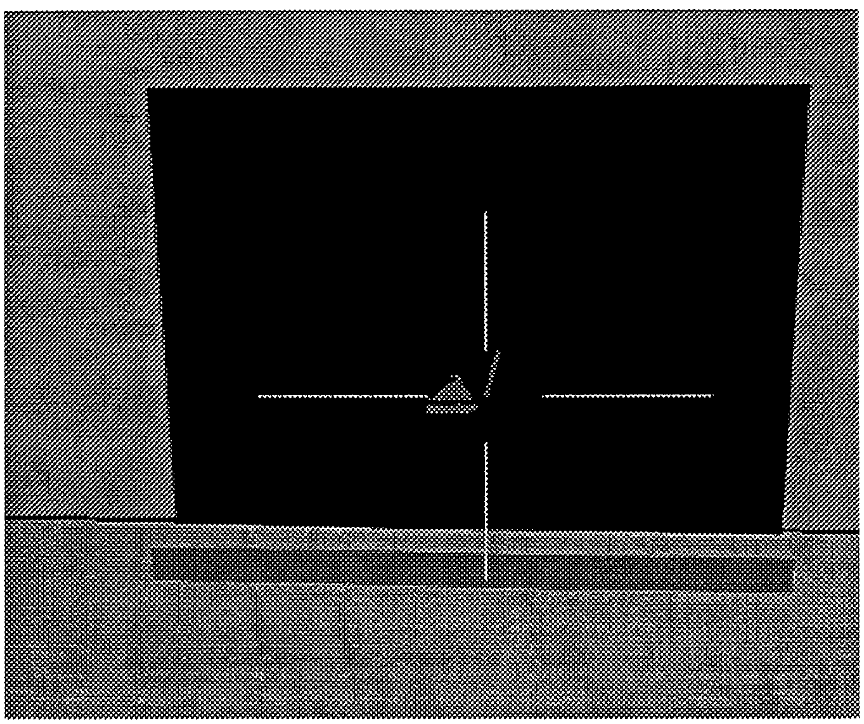

Figure 25: Chief of the Watch Station

The remaining area of control includes six display consoles. These consoles are non-functional and provided only to provide some of the "look and feel" of an actual control room. Figure 26 shows a view of the control room from the conning station.

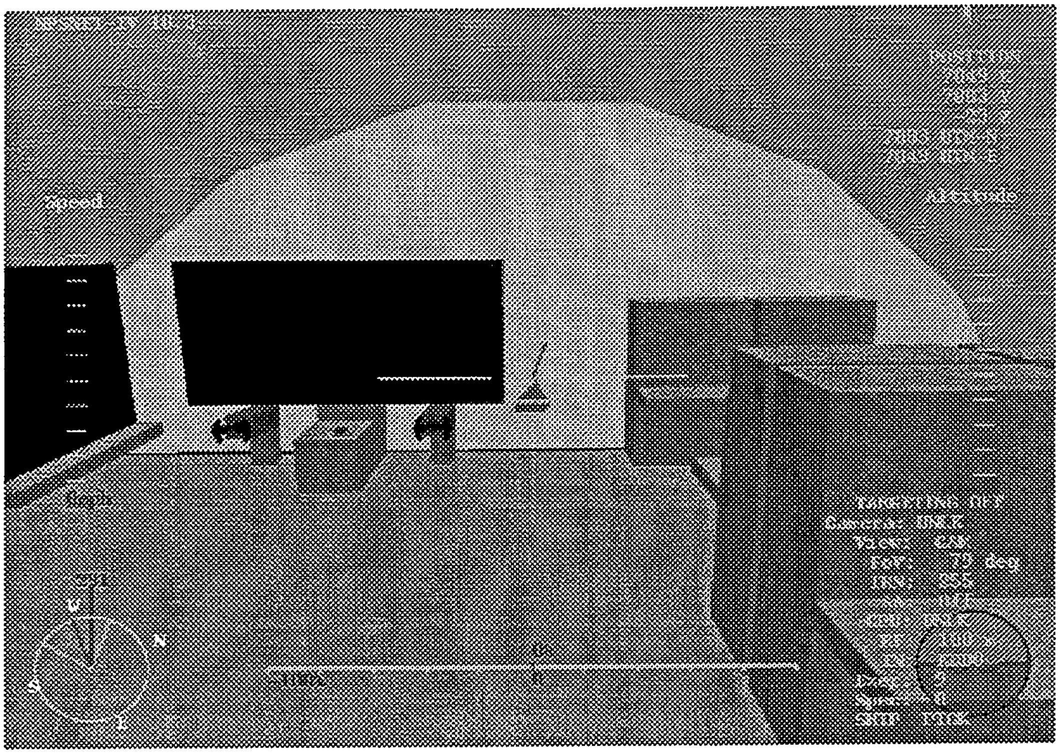

Figure 26: OOD's view of Control Room from Conning Station

C. MOUNTING ALGORITHM

The NPSNET ship simulator provided a new paradigm for interacting with a virtual shipboard environment by mounting human entities to it [Stewart 96]. This serves as a good basis for mounting human entities on a submarine; however, there are several characteristics inherent to a submarine that necessitate further design effort.

Human entities in NPSNET are implemented as an object-oriented hierarchy of C++ classes. Progressing downwards in the hierarchy, the behavior of each class becomes more specific to the particular type of human entity. A new class, jack_submariner_vehicle was added to encapsulate the behavior specific to human entities mounted to submersible entities. Figure 27 depicts the revised class hierarchy as a result of this work.

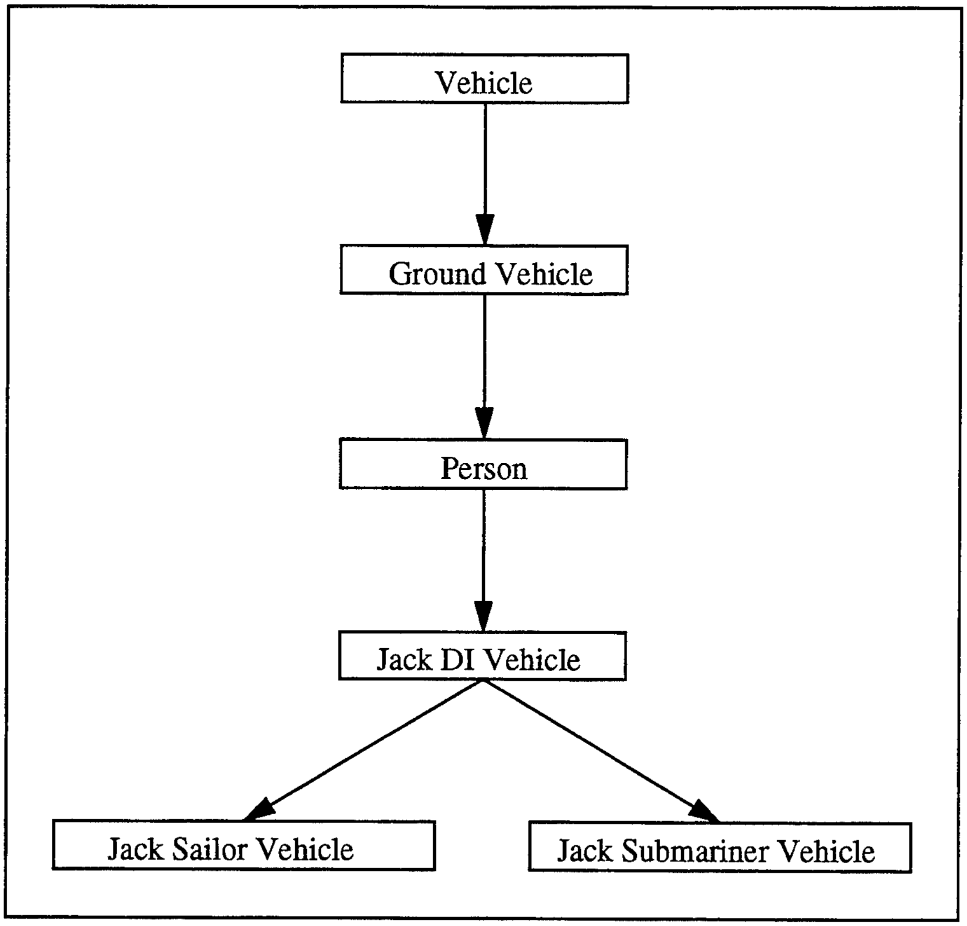

Figure 27: Class Hierarchy for Human Entities in NPSNET

1. Maintaining Correct Position of Mounted Human Entities

In the submarine simulator, a human entity mounts a submarine when it is determined that the bounding volumes of the human and a part of the submarine have intersected. Specifically, a test is performed to determine if a vector, pointed downwards in the vertical direction (negative z axis) intersects a part of the ship such as the control room deck. Once mounted, the human becomes an articulated part of the submarine, positioned relative to the submarine's position (local coordinates). Unfortunately, current DIS standards support only global positions. In order to correctly position the human entity, a global position must be determined based on the mounted submarine's position, and the position of the human in local coordinates (Figure 28 depicts this in the xy plane).

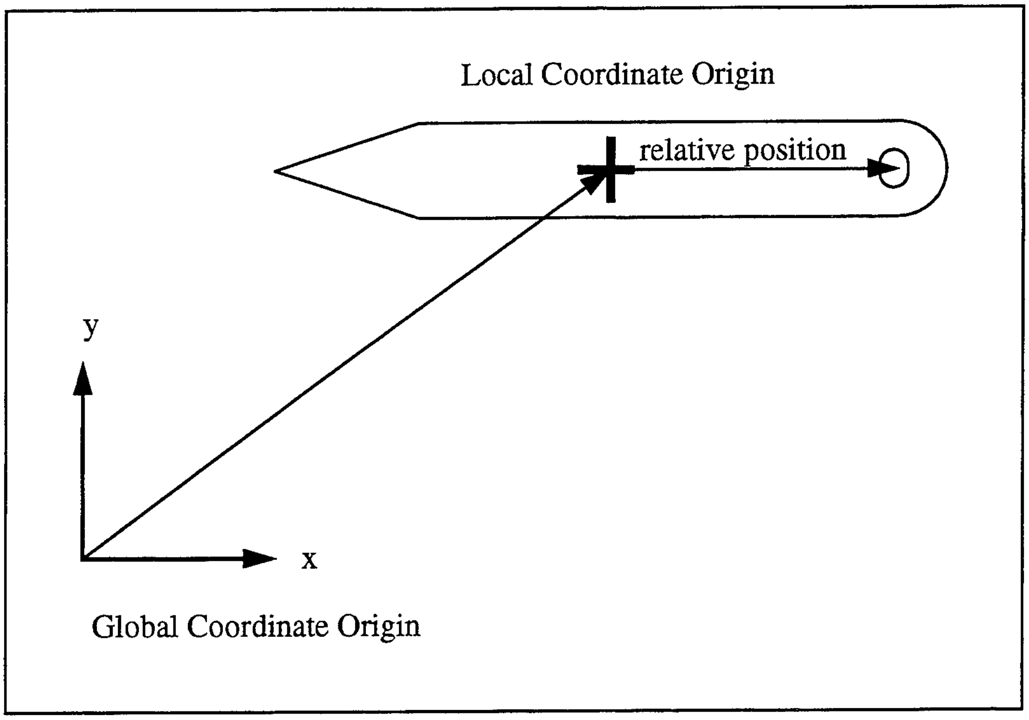

Figure 28: Calculating Mounted Human Entity Global Position

When the bounding boxes of a human entity and a part of the submarine intersect in the NPSNET submarine simulator, an initial relative position vector is calculated based on the global positions of the two entities. During each frame of the simulation, the human entity first calculates its updated position as if the submarine were a static object, accounting for the human entity's movement about the submarine. Then it checks for collisions with the submarine's bulkheads and decks before determining an intended position. After finalizing its intended position, the relative position is updated from the new position of the human entity. After the mounted submarine entity has completed its movement for the frame, the updated relative position vector is computed. The relative position vector is then rotated to account for the change in orientation of the submarine. Unlike mounted human entities in the ship simulator, the pitch and roll angles of the submarine entity are taken into account for this rotation. The relative position vector is then

added to the mounted submarine's global position to determine the new global position of the mounted human entity.

2. Maintaining Correct Position of Networked Ghost Human Entities

Since the networked ghost human entities rely on Entity State Update PDUs (ESPDUs) and dead reckoning to maintain position, a different approach is needed since these PDUs are not received every frame. As a result, without another approach, the ghost human entities are always out of position between the receipt of PDUs when the mounted submarine entity is moving. Although the submarine may be moving along at some speed, the ghost human entities have a speed of zero, thus they are continually snapping back into position when Heartbeat PDUs are received. One approach is to send ESPDUs every frame but this can quickly saturate the network if several humans are mounted to the submarine. Another approach is to inform the ghost entities that they are mounted to a submarine. This is the approach that was pursued.

D. HIGH RESOLUTION NETWORK

The High Resolution Network (HIRESNET) is utilized by NPSNET to communicate information that is not supported by the DIS standard. The HIRESNET uses the same type of net manager for both DIS network and HIRESNET communication. The HIRESNET is assigned a different multicast port. The HIRESNET provides the Data PDU which allows the communication of some of the data not supported by the DIS standard. Table 1 delineates the data fields of the Data PDU. Only those entities listening to the HIRESNET actually process the traffic from this network.

Currently the HIRESNET is used to send specific human arm joint positions across the network. The information is more detailed than the articulations in the ESPDU [Stewart 96]. The HIRESNET is used to support two requirements for the submarine simulator. First, it is used to send information on manipulated objects such as periscopes, control surfaces, and the Engine Order Telegraph to ghost submarine entities. This information is

utilized by the ghost submarine entities to properly position these objects. Secondly, it is used to inform ghost human entities that they are mounted to a submarine entity.

Field Size (bits)	HIRESNET Data PDU Fields	
96	PDU Header	The information needed by any receiving node to properly characterize and operate on the data being received
48	Originating Entity ID	ID and force of originating entity
8	Originating Force ID	
48	Receiving Entity ID	ID and force of receiving entity
8	Receiving Force ID	
8	Request ID	Type of data in datum records
8	Num_datum_fixed	Total number of Datum Records
n * 32	Fixed_datum	Datum Records
Total PDU size = (224 + 32n) bits where n = number of fixed datum records		

Table 1: HIRESNET Data PDU Structure [Stewart 96]

In a manner similar to the NPSNET ship simulator, the submarine simulator uses the HIRESNET to relay mounting information to ghost human entities [Stewart 96]. When a human entity mounts a submarine entity, a special mounting identification tag is placed in the Request ID field of a Data PDU and the entity id of the mounted submarine entity is placed in the Receiving Entity ID field of the PDU. The PDU is then sent and when a ghost human entity receives the PDU it mounts the submarine as well, storing the entity state id of the submarine and setting a relative position vector. In addition, when the HIRESNET manager running on the same workstation as the ghost human entity sees the special mounting tag, a function is executed in the submersible vehicle class to add the ghost human entity to an array of pointers to mounted human entities. Then during each frame when the submarine entity dead reckons, a function is called in the jack_submariner_veh

class to update the relative position vector and reset the position of all mounted human entities.

E. SUMMARY

The NPSNET ship simulator proved to be a useful paradigm for the mounting of human entities to another entity in the virtual environment. The relative position between the human and ship entities is used to compute the updated global position for the human entity each frame. The HIRESNET is utilized to inform ghost human entities to mount the ship. Once the ghost entities mount the ship, their positions are correctly computed each frame. This eliminates the snapping of the the human entity's position that occurred because the Entity State PDU position data varied significantly from the current position of the ghost entity. The human entity class hierarchy has been expanded to include a new class, jack_submariner_veh that implements a similar algorithm for human entities mounted to submarine entities. However, changes have been made to account for pitch and roll angles of the submarine. In addition the intersection tests with the terrain have been removed since these human entities are intended to be placed in the submarine's control room, which is currently an enclosed volume. There is no need to walk up a ramp or other object to mount the submarine.

In order to mount the human entities to the submarine, a control room was added to the Multi Gen OpenFlight submarine model. The positioning of this control room within the model is representative of the position of a control room on an actual submarine. Several manipulable objects were added to this control room including periscope hand wheels, forward and stern plane control sticks, rudder steering wheel, and Engine Order Telegraph.

As a result, human entities can mount and control and maneuver a submarine entity in the NPSNET virtual environment. A group of human entities mounted to the submarine entity can act together as a watch team representing members of the ship's control party.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. RESULTS

The goal of this research was to improve the interface and control of the NPSNET submarine simulator. To achieve that goal, this thesis explores network mounting of a group of human entities to the submarine who act together as a watch team, controlling and maneuvering the submarine entity in the virtual world. In addition another interface was needed which allowed a single user to control and manipulate the submarine entity. The following results have been achieved as a result of this work:

1. MultiGen OpenFlight Submarine Model

The MultiGen OpenFlight visual submarine model has been expanded by adding several Degree of Freedom (DOF) nodes to the database. With these DOF nodes the following objects can be manipulated by the application:

- Four DOF nodes are provided for the port and starboard periscopes so that the housings, fairings, scopes and eyepieces can be moved. This allows for raising and lowering the periscope as well as rotating the eyepiece to change the bearing and altitude of the scope eyepiece.

- A single DOF node is provided for each control surface (rudder, forward planes, stern planes) to allow for rotation as the user changes rudder and planes angles.

- Three DOF nodes are provided for both the Helm station and Dive Planesman station. These control sticks consist of vertical and horizontal linkages, as well as a steering wheel. The vertical linkage may be rotated, and the horizontal linkage and steering wheel translated to simulate the operation of an actual control stick. The steering wheel may be rotated to simulated the operation of a rudder steering wheel. The steering wheel on the Dive Planesman control stick does not rotate.

- A DOF node is provided for the Engine Order Telegraph (EOT). This allows the EOT to rotate, thus indicating the currently ordered bell.

- A DOF node is provided for each periscope operating hand wheel. This allows the handwheel to rotate as the user selects it to raise or lower the periscope.

2. The Single User Interface

A remote control is provided to operate the simulator in single user mode. The user can change the submarine's speed (bell), rudder and planes angles, periscope positions (raised or lowered), the current viewpoint (from a raised periscope or control), and the bearing and altitude of a periscope viewpoint.

3. The Watch Team Interface

A group of human entities can be mounted to the submarine to act together as a watch team. Watch stations are provided in the control room for the Officer of the Deck, Diving Officer of the Watch, Chief of the Watch, Helmsman/Planesman, and Dive Planesman. At some of the watch stations are objects that the user can "pick" by positioning an icon and pressing the left or right mouse buttons. The following objects can be picked:

- The periscope handwheels can be picked by the Officer of the Deck. By picking a handwheel, the associated periscope either raises or lowers depending on the initial position.
- At the Helm Station, picking either the vertical or horizontal linkage of the control stick moves the control stick as well as the forward planes. Picking either linkage with the left mouse button causes the vertical linkage to rotate forward and the horizontal linkage and steering wheel to translate forward and downward and the forward planes to rotate in the dive direction. Conversely, picking the linkages with the right mouse button causes the control stick to move aft and the forward planes to rotate in the rise direction. The new forward planes angle is then inputted to the hydrodynamics model. Picking the steering wheel with the left mouse button causes it to rotate to the left and the rudder to move left. Picking the steering wheel with the right mouse button causes it to rotate right and the rudder to move right. The new rudder angle is then inputted to the hydrodynamics model.
- At the Helm Station, picking the EOT with the left mouse button causes the EOT to rotate to the left and the ship's bell to be decreased. Conversely picking the EOT with the right mouse button causes the EOT to rotate to the right, and the ship's bell to be increased. The new bell is then inputted to the hydrodynamics model.
- At the Dive Planesman station, the operation of the control stick is similar to the operation of the control stick at the Helm Station. The only difference is that the stern planes angle changes vice the forward planes angle. Also, picking the steering wheel on this control stick performs no action.

4. Communications

Data PDUs from the HIRESNET are used to inform ghost entities that they are mounted to submersible entities. A special mounting tag placed in a data field of the PDU triggers the mounting. Once mounted the ghost entities are updated each frame based on the mounted submarine entity's position and a relative position vector between the ghost human entity and the mounted submarine entity. The HIRESNET is also utilized to update movable objects on ghost submarine entities such as periscopes and ship's control surfaces.

5. Working with NPSNET

Making revisions to the current version of NPSNET (NPSNET IV-10.9) has reached the point of diminishing returns. As new features have been added to NPSNET, there has been no effort to redesign the system as a whole. The current version is a hybrid mixture of the object-oriented, and non-object-oriented paradigms (sometimes within the same file). As a result, it took approximately six months to learn the basics of NPSNET such as communications, entity updates, etc. In addition, it is very difficult to determine what the behavior of the system will be when changes are made to a specific application such as the submarine simulator. Even though human entities were previously mounted to a ship in the ship simulator [Stewart 96], applying the same techniques for mounting humans to a submarine entity was very difficult due to the lack of a "tool kit" which ideally would make this a relatively simple implementation.

B. RECOMMENDATIONS FOR FUTURE WORK

This thesis work provided a major improvement to the realism and immersion qualities of the NPSNET submarine simulator. It extended and improved on previous work from the NPSNET submarine and ship simulators. More research and development is needed to make the submarine simulator a viable training tool for the United States Submarine Fleet. Specifically, more work is needed in the following areas:

- Build a VRML version of the NPSNET submarine simulator with networking and dynamics implemented using Java.

- Integrate interactive physically based ship models and real-time 3D graphics with shipboard combat control systems and afloat/ashore training systems.
- Obtaining or developing sets of hydrodynamic coefficients for both the LOS ANGELES and TRIDENT class submarines for use with the existing hydrodynamics model.
- Extend the capabilities of the submarine simulator by adding weapons such as Torpedoes and Missiles. The user should be able to easily launch these weapons from a realistic interface. The user should be able to specify initial parameters for the weapon such as course. The weapons should exhibit autonomous behavior and the motion of the weapons should be physically based.
- Develop dynamic collision detection for NPSNET. A collision-detection algorithm should be used which does not impede performance while providing the detail necessary to detect exact collision points between objects.
- Develop a more intuitive picking mechanism. By utilizing the existing motion library for human entities, extend the picking mechanism so that an object can be picked by placing a finger on it.
- Explore the use of haptic feedback devices to allow the user to manipulate various objects in the virtual world, while at the time experiencing the touch and feel of actually manipulating these devices.
- Obtain actual CAD data, if possible to provide an accurate visual model of both the exterior and interior areas of the LOS ANGELES and TRIDENT class submarines. Expand the model to include all interior areas of the submarine.
- Add dynamic casualties to the simulation such as fires and flooding.
- Add a graphical user interface to NPSNET for starting all applications available in NPSNET. The GUI should allow the user to change various parameters for the simulation such as starting position and orientation.
- Extend the existing ocean wave carpet model to provide input to the hydrodynamics model to account for current induced forces on the submarine's hull surfaces.
- Expand communications to include a voice channel so that orders can be given and acknowledged between watchstanders over the network. This would include a natural voice recognition system as well as playback capabilities at each workstation using public domain multicast audio software.

APPENDIX A. MODEL HIERARCHY

The NPSNET submarine simulator utilizes a three dimensional (3D) model of a Los Angeles class submarine which was designed using Multi Gen. Multi Gen is a 3D modeling tool which allows the designer to specify a database hierarchy for the model. The user can build a hierarchical tree structure of nodes. The types of nodes (referred to as beads in Multi Gen) used in this model include:

- Group nodes allow the user to specify two data fields to be stored with the node (special1 and special2). Group nodes can have any number of children, including other group nodes.
- Degree of freedom (DOF) nodes allow the user to specify rotations and translations that are applied to all nodes under the DOF node. DOF nodes must have a group node as a parent and may have either group nodes or object nodes as children.
- Object nodes also allow the user to specify two data fields to be stored with the node. Object nodes must have a group node as their parent and may only have face nodes (or polygons) as children.
- Face nodes (polygons) are the basic building blocks of the model. Face nodes must have either a group node or object node as their parent. Faces may have sub-faces as children. Each face is specified by a set of vertices that describe points in three dimensional space.

These node types along with some other types of nodes not described here allow the user to build a 3D model of an object such as a submarine, tank, or ship storing the model in a database hierarchy which is written as an Open Flight (.flt) file. Various loaders are available which will read the Open Flight format. In NPSNET, the Open Flight loader included with Performer 2.0 is utilized to perform this task.

Figures 29 through 40 outline the database hierarchy of the submarine model used by NPSNET. The special1 and special2 fields are used at load time to identify certain group nodes with DOF nodes as children or PVS cells. Pointers are set to these nodes during the database loading phase for later manipulation in the simulation. Tables 2 and 3 summarize the data contained in the special1 and special2 fields of the model's group nodes. The special1 and special2 fields are not used in any of the object nodes in this model.

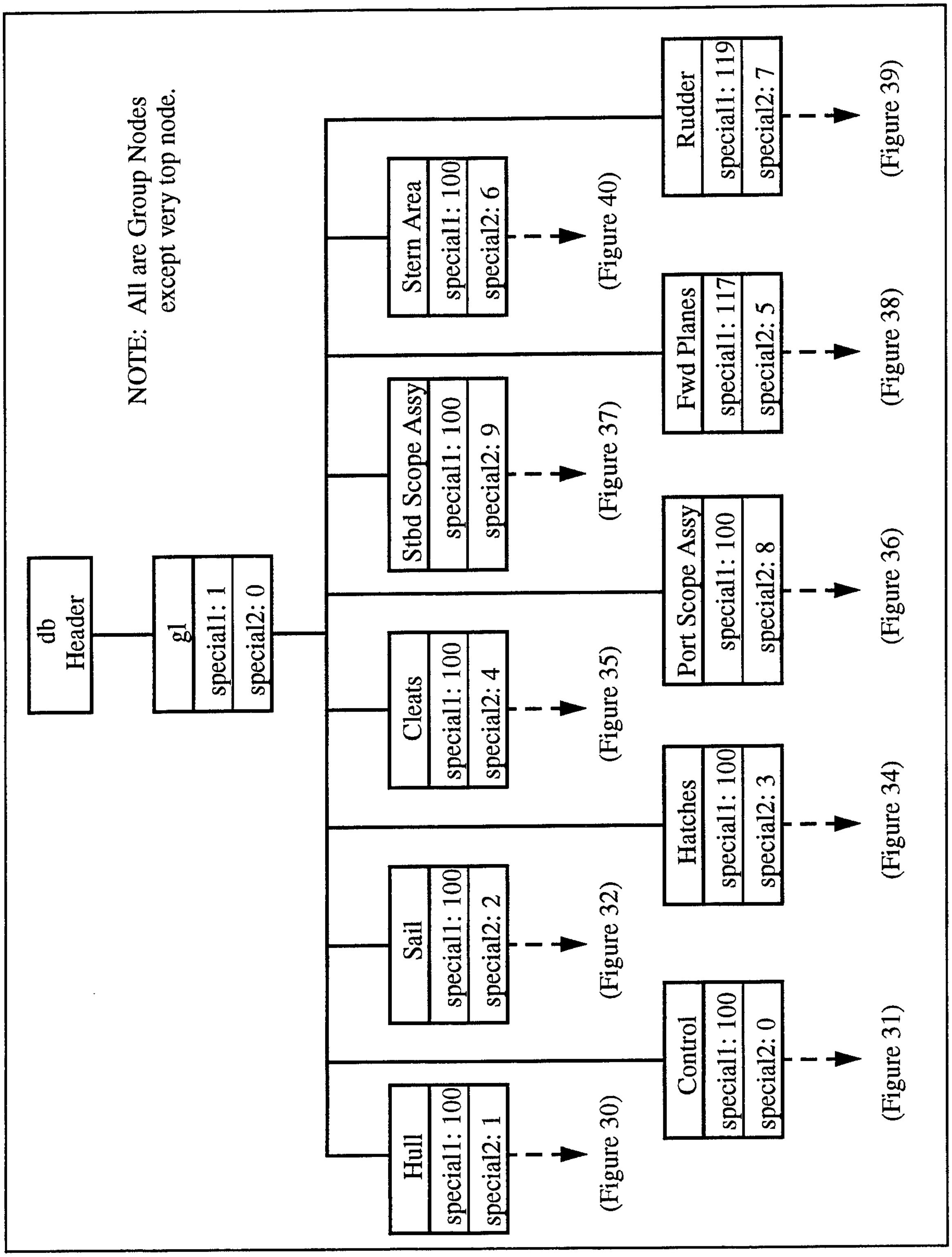

Figure 29: Upper Level of Submarine Model Hierarchy

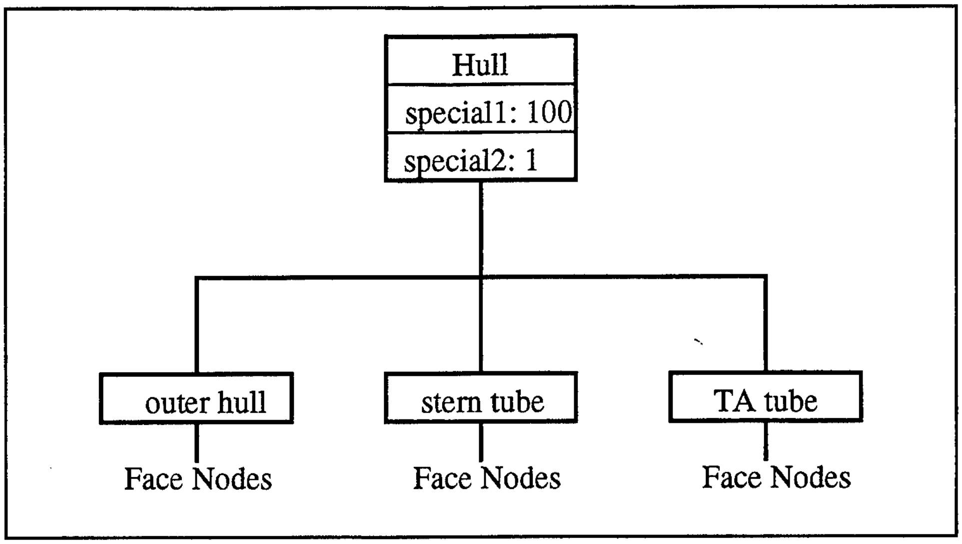

Figure 30: Hull Group Node and below of Submarine Model Hierarchy

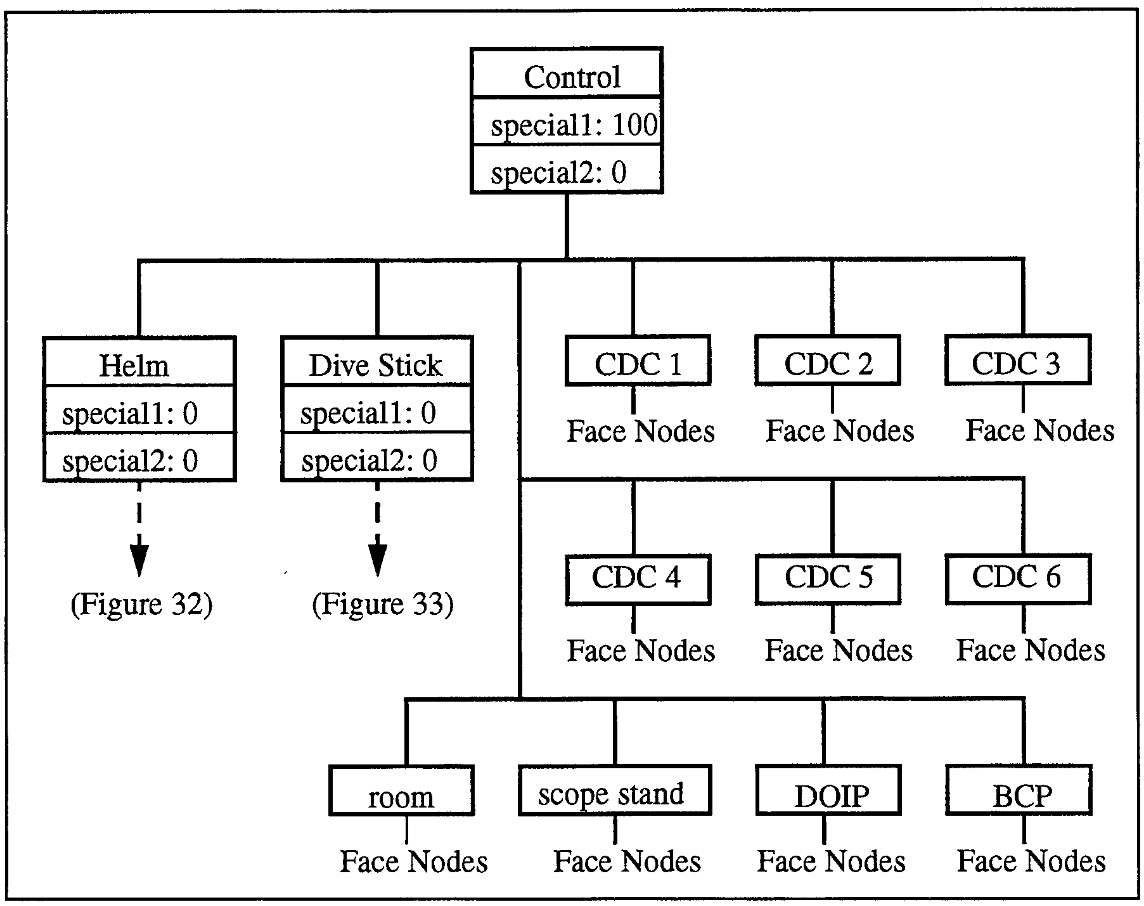

Figure 31: Control Group Node and below of Submarine Model Hierarchy

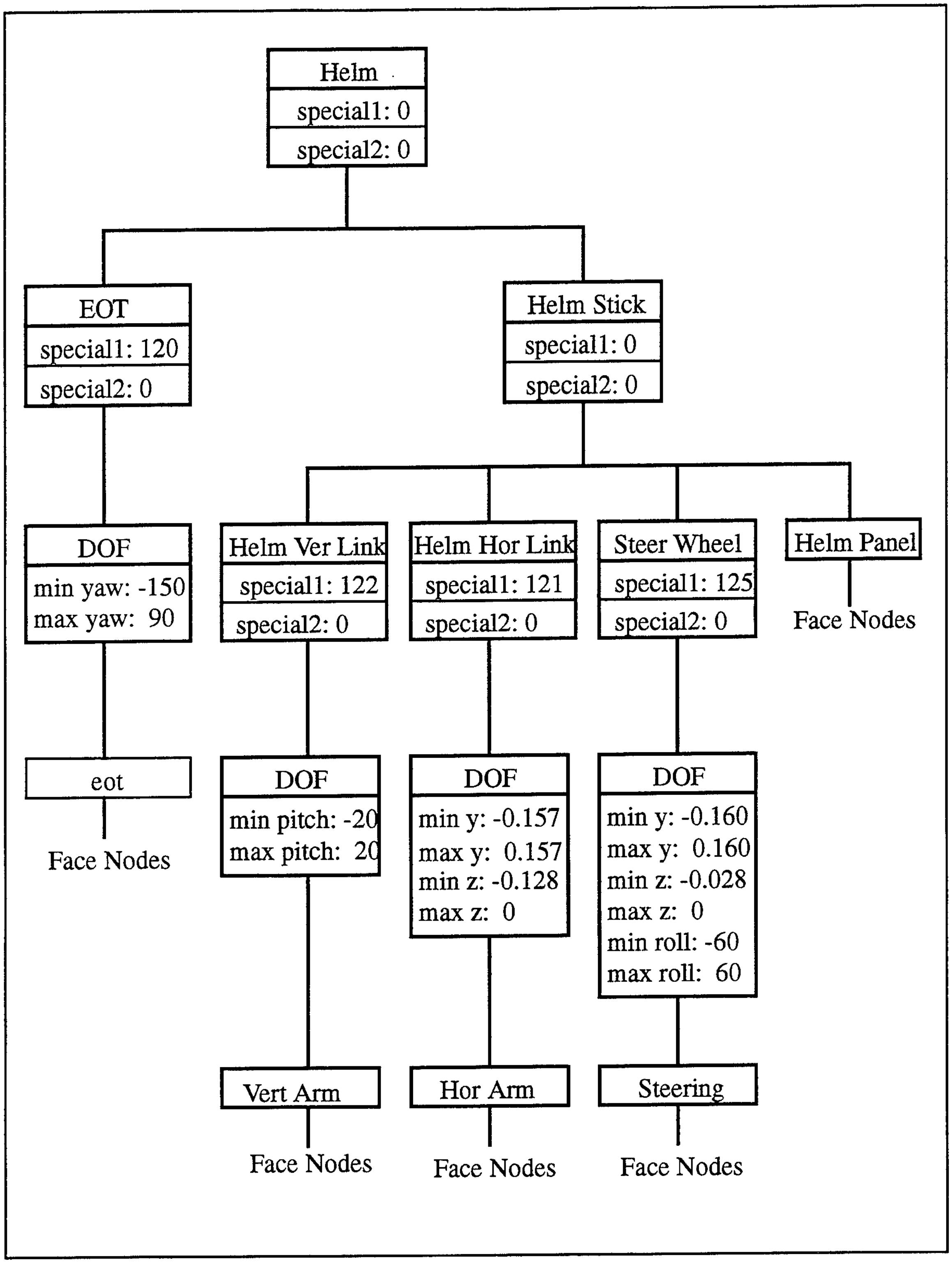

Figure 32: Helm Group Node and below of Submarine Model Hierarchy

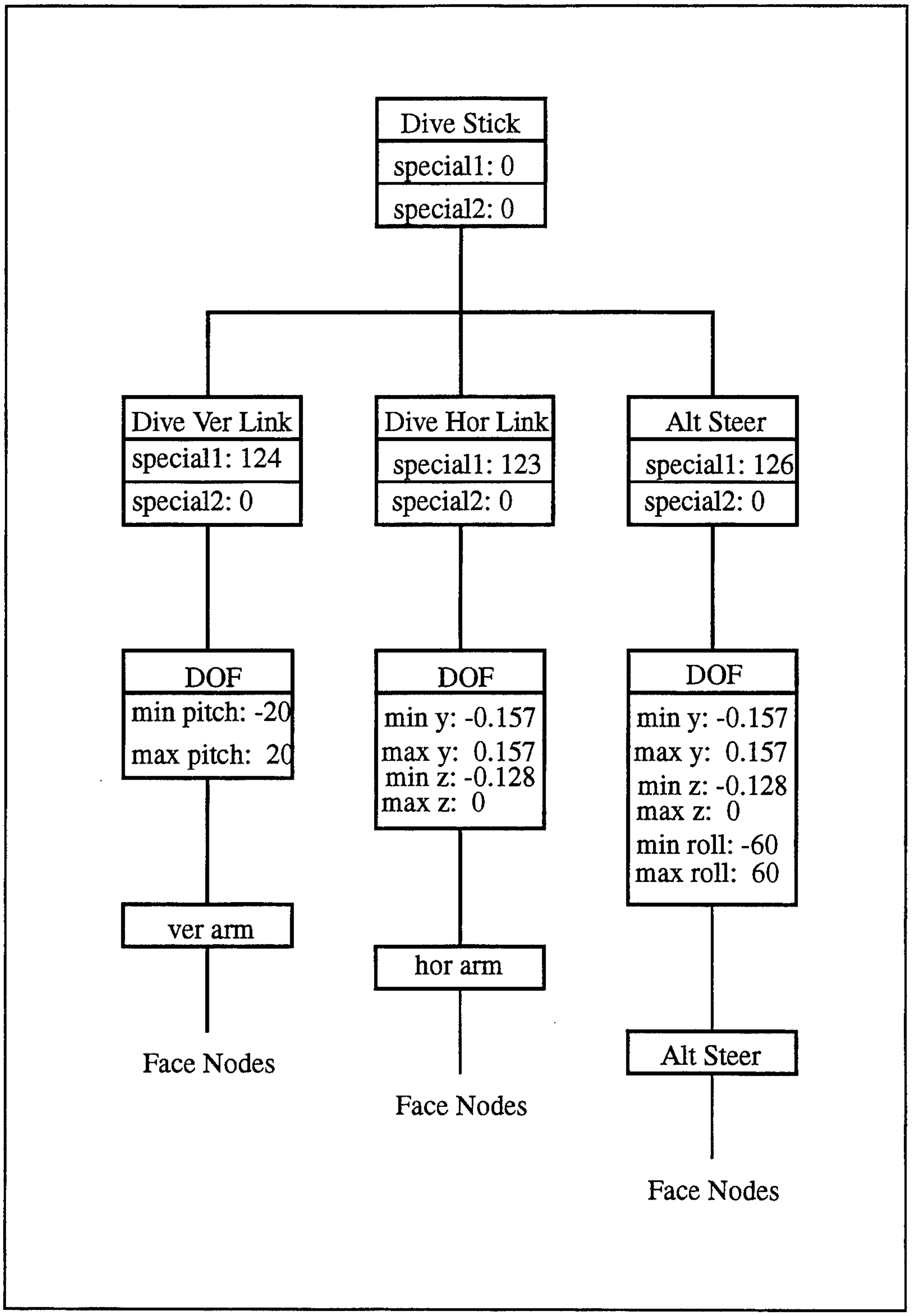

Figure 33: Dive Stick Group Node and below of Submarine Model Hierarchy

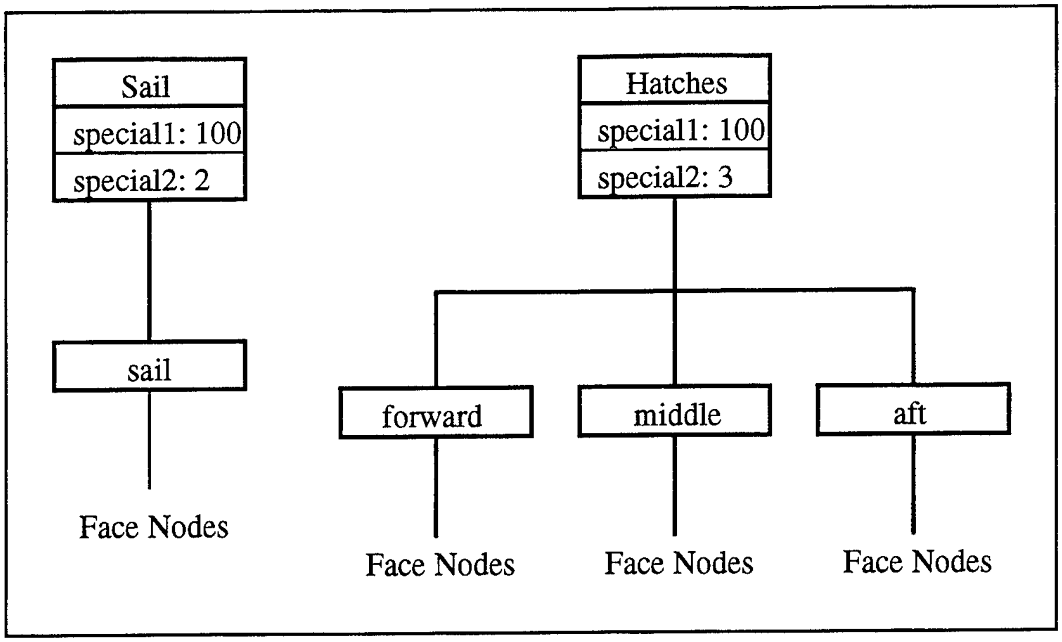

Figure 34: Sail and Hatch Group Nodes and below of Submarine Model Hierarchy

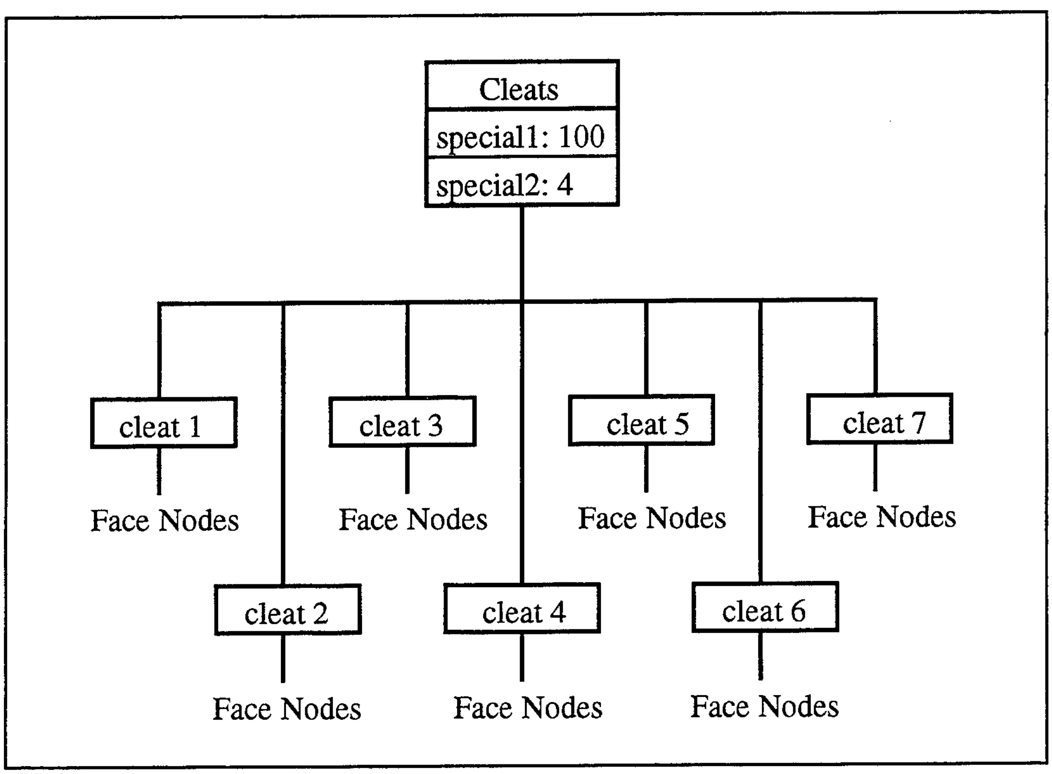

Figure 35: Cleats Group Node and below of Submarine Model Hierarchy

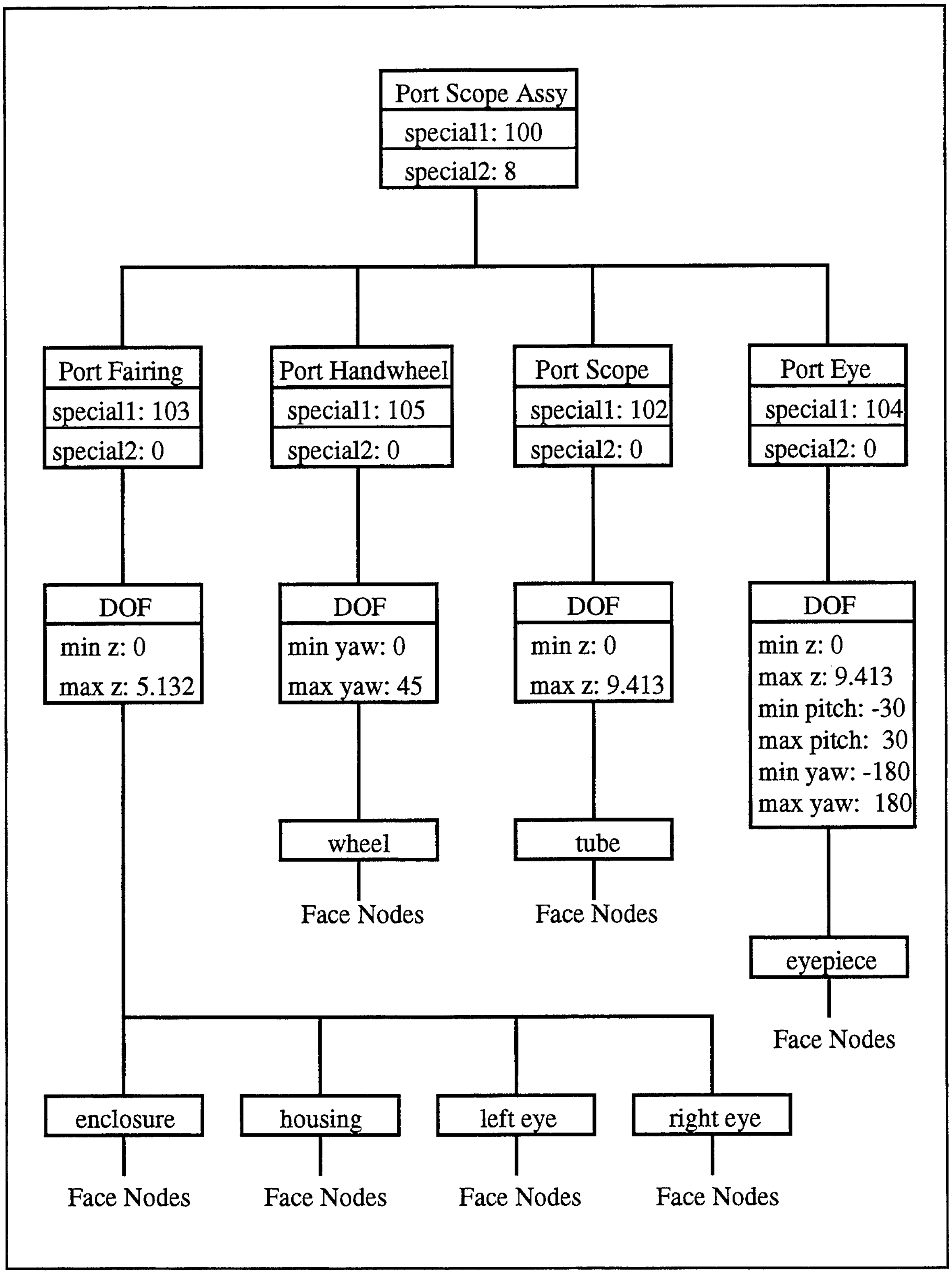

Figure 36: Port Scope Assy Node and below of Submarine Model Hierarchy

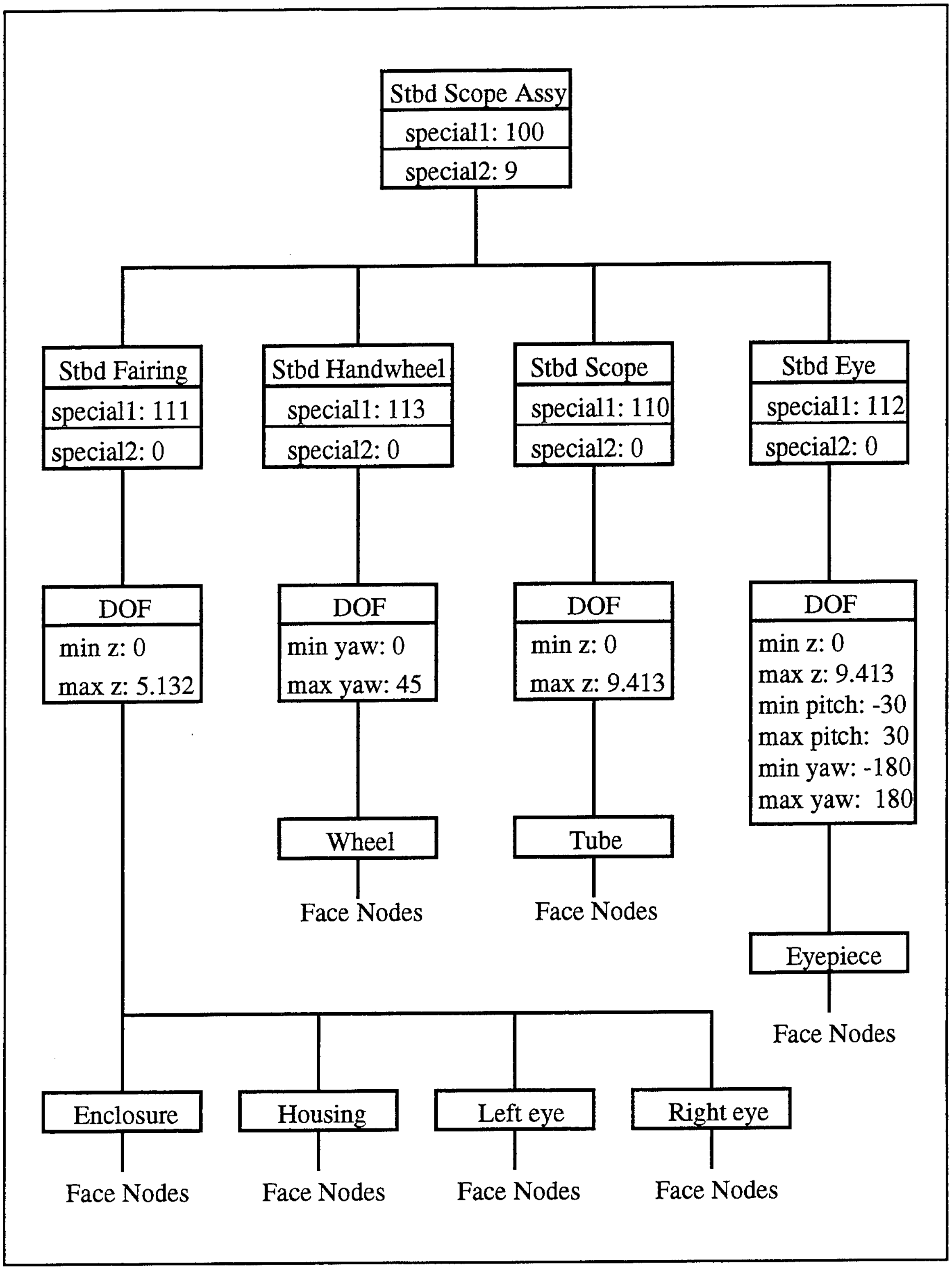

Figure 37: Stbd Scope Group Node and below of Submarine Model Hierarchy

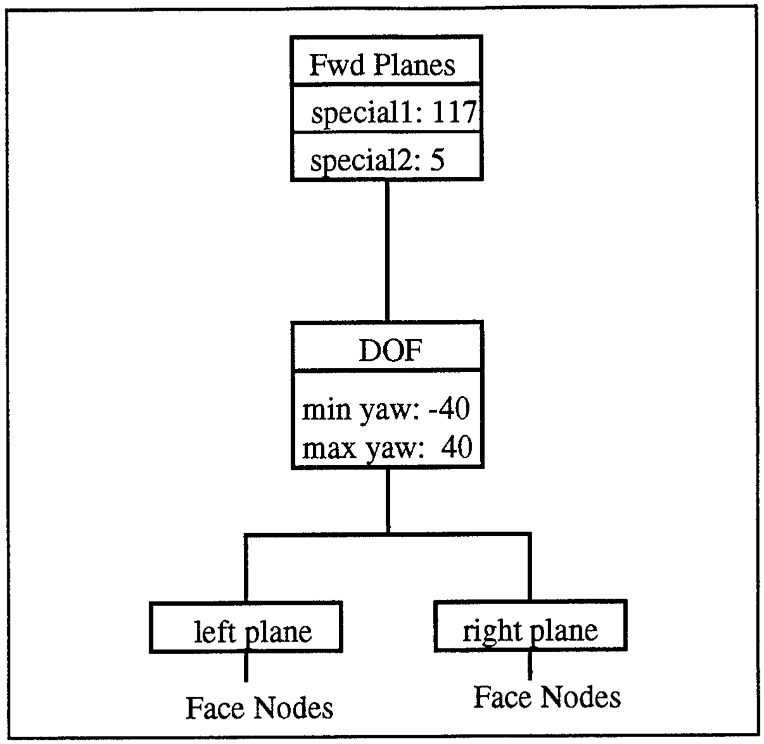

Figure 38: Fwd Planes Group Node and below of Submarine Model Hierarchy

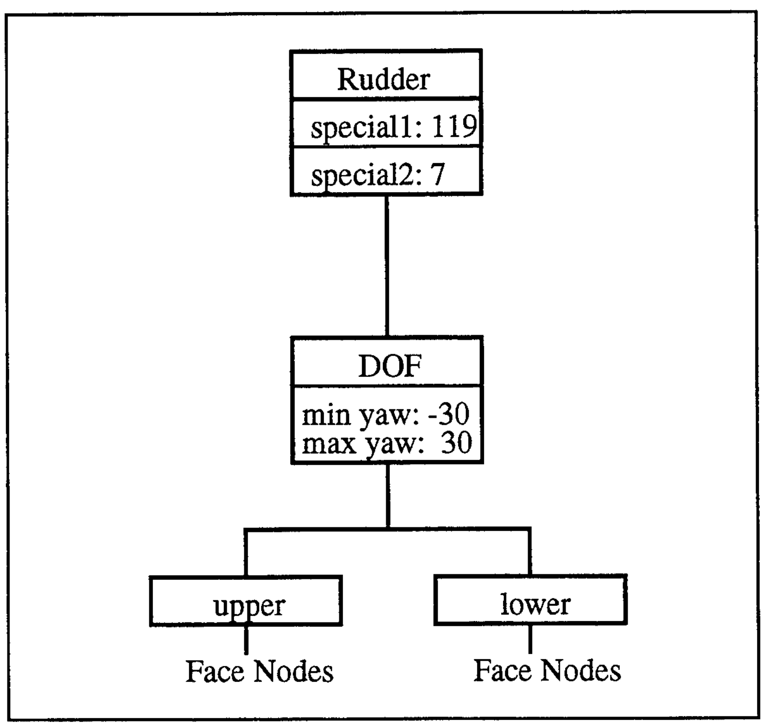

Figure 39: Rudder Group node and below of Submarine Model Hierarchy

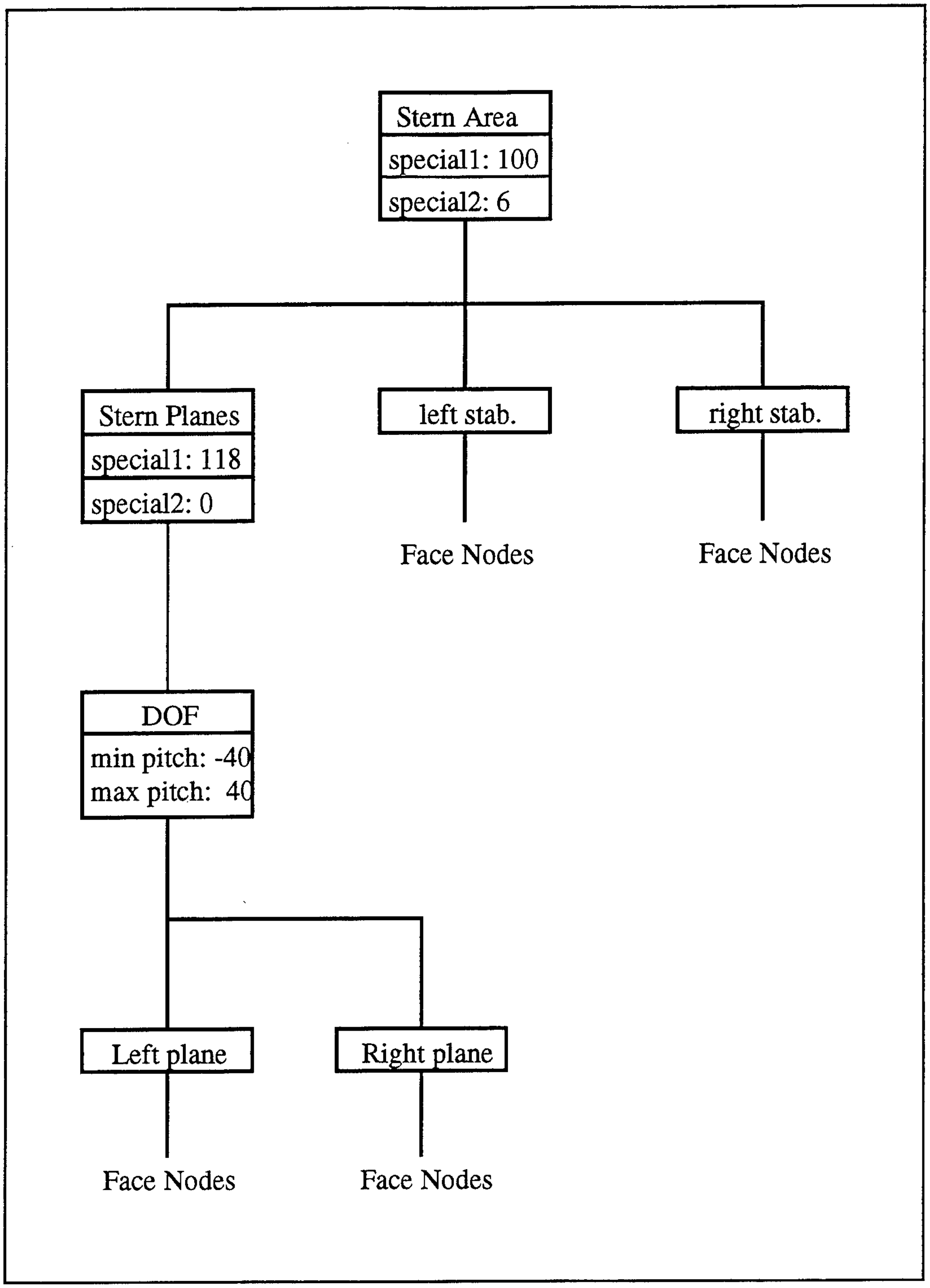

Figure 40: Stern Area Group Node and below of Submarine Model Hierarchy

Table 2: Summary of Group Node Types

special1 value	Node Name	Purpose
100	Several Nodes	PVS Cells (see table 3)
102	Port Scope	Allows the port periscope to be translated (z).
103	Port Fairing	Allows the port periscope fairing and housing to be translated (z).
104	Port Eye	Allows the port periscope eyepiece to be translated (z), and rotated (pitch and yaw).
105	Port Handwheel	Allows the port periscope operating handwheel to be rotated (yaw).
110	Stbd Scope	Allows the starboard periscope to be translated (z).
111	Stbd Fairing	Allows the starboard periscope fairing and housing to be translated (z)
112	Stbd Eye	Allows the starboard periscope eyepiece to be translated (z), and rotated (pitch and yaw).
113	Stbd Handwheel	Allows the starboard periscope operating handwheel to be rotated (yaw)
117	Fwd Planes	Allows the forward planes (fairwater) to be rotated (yaw).
118	Stern Planes	Allows the stern planes to be rotated (pitch).
119	Rudder	Allows the rudder to rotated (yaw).
120	EOT	Allows Engine Order Telegraph to rotate when picked by user to change ordered bell
121	Helm Hor Link	Allows horizontal linkage on the control stick at the Helm Station to rotated (pitch).
122	Helm Ver Link	Allows vertical linkage on control stick at the Helm Station to be translated (y and z).
123	Dive Hor Link	Allow horizontal linkage on the control stick at the Dive Planesman Station to be translated (y and z).
124	Dive Ver Link	Allows vertical linkage on control stick at the Dive Planesman Station to be rotated (pitch)

special1 value	Node Name	Purpose
125	Steer Wheel	Allows steering wheel on control stick at Helm Station to be rotated (roll) and translated (y and z).
126	Alt Steer	Allows steering wheel on control stick at Dive Planesman station to be translated (y and z).

Table 3: Summary of PVS Cells

PVS Cell (special2 value)	Group Node	Cells potentially visible from this cell.	Description
0	Control	0, 8, 9	The control area of the submarine model.
1	Hull	1	The hull surface of the submarine.
2	Sail	2	The sail of the submarine
3	Hatches	3	The topside hatches of the submarine.
4	Cleats	4	The mooring cleats for the submarine.
5	Fwd Planes	5	The forward (fairwater) planes of the submarine
6	Stern Area	6	The stern planes and stabilizers of the submarine
7	Rudder	7	The rudder of the submarine.
8	Port Scope Assy	8	The parts of the port periscope assembly including fairing, housing, periscope tube and eyepiece.
9	Stbd Scope Assy	9	The parts of the starboard periscope assembly including fairing, housing, periscope tube and eyepiece.

PVS Cell (special2 value)	Group Node	Cells potentially visible from this cell.	Description
10	None	1, 2, 3, 4, 5, 6, 7, 8, 9	The default PVS cell. When the current viewpoint does not fall within the bounding volume of any other cell, all exterior surfaces of the submarine are potentially visible.

APPENDIX B. VIDEO DEMONSTRATION

A. INTRODUCTION

Briefly describes the importance of developing viable training tools to train junior submarine officers in the art and science of shiphandling.

B. SINGLE USER INTERFACE

This segment shows the NPSNET submarine simulator operating in single user mode. Demonstrates the use of the features on the submarine control panel. Specifically, the following are demonstrated (not necessarily in this order):

- Raising and lowering of periscopes (as seen from inside the control room, and from outside the ship).
- Selecting the current viewpoint (port scope, starboard scope, or control).
- Changing the bearing and altitude of a scope view point.
- Increasing and decreasing ordered bells.
- Placing rise angles on the forward and aft planes to surface the submarine.
- Placing dive angles on the forward and aft planes to submerge the submarine.
- Changing ship's course with rudder angles changes.

C. MOUNTED HUMAN ENTITY INTERFACE

This segment shows a group of human entities mounted to the submarine, acting together as a watch team. Demonstrates the use of the picking mechanism to perform the following actions:

- Raise and lower periscopes by picking the appropriate orange hand wheel (OOD).
- Change propulsion orders by picking the Engine Order Telegraph (Helm). Show both decreasing and increasing bell orders.
- Change both forward and stern planes by picking control stick linkages at both the Helm and Dive Planesman stations.
- Show course changes by picking the steering wheel at the Helm station.

APPENDIX C. OBTAINING NPSNET SOFTWARE AND DOCUMENTS

A. OBTAINING NPSNET SOFTWARE

NPSNET is currently undergoing a major revision. This revision entails a major ground up design effort to completely build a new architecture for the system. This will be released as NPSNET V sometime in the near future. The current version of NPSNET, NPSNET IV-10.3 is available at the following web site as either source code files or executable files:

http://www-npsnet.cs.nps.navy.mil/npsnet/register.html

Since version IV-10.3 of NPSNET does not include the changes made to the NPSNET submarine simulator as a result of the work of this thesis, a patch for rmounting human entities to a submarine entity in NPSNET is also available at the above web site. With NPSNET version IV-10.3 and this patch, the revised submarine simulator is provided. The revised submarine simulator may be included as a part of NPSNET V or a later revision.

B. OBTAINING NSPNET DOCUMENTS

Various documents related to the development of NPSNET and other subjects related to large scale virtual environments are available on line. This thesis, along with several others produced by members of the NPSNET Research Group are also available on line. These documents can be obtained from the following web site:

http://www-npsnet.cs.nps.navy.mil/npsnet/publications.html

LIST OF REFERENCES

[Bacon 95] Bacon, Daniel K. Jr., *Integration of a Submarine into NPSNET,* Master's Thesis, Naval Postgraduate School, Monterey California, September 1995. Available at *http://www-npsnet.cs.nps.navy.mil/npsnet/publications.html*

[Barham 94] Barham, Paul T., Zyda, Michael J., Pratt, David R., Lock, John, Falby, John, *NPSNET_IV: A DIS-Compatible, Object-Oriented Software Architecture for Virtual Environments*, unpublished paper, Naval Postgraduate School, Monterey California, October 1994.

[Brutzman 94] Brutzman, Donald P., *A Virtual World for an Autonomous Underwater Vehicle*, Ph.D Dissertation, Naval Postgraduate School, Monterey California, March 1994. Available at *http://www.stl.nps.navy.mil/~brutzman/dissertation*

[Noble 95] Nobles, Joseph and Garrova, James, *Design and Implementation of Real-Time, Deployable Three Dimensional Shiphandling Training Simulator,* Master's Thesis, Naval Postgraduate School, Monterey California June 1995. Available at *http://www-npsnet.cs.nps.navy.mil/npsnet/publications.html*

[NAWC 96] Naval Air Warfare Center Training Systems Division, Special Report 96-002, *Virtual Environment Submarine Officer of the Deck Simulation and Training: Preliminary Requirements Recommendation*, Orlando Florida, June 1996.

[Obyrne 95] O'Byrne, James E., *Human Interaction within a Virtual Environment for Shipboard Training*, Master's Thesis, Naval Postgraduate School, Monterey California, September 1995. Available at *http://www-npsnet.cs.nps.navy.mil/npsnet/publications.html*

[Pioch 95] Pioch, N., *Officer of the Deck: Validation and Verification of a Virtual Environment for Training*, Research Labroatory of Electronics, Massachusetts Institute of Technology (MIT), Cambridge Massachusetts, June 1995.

[SGIC 94] Silicon Graphics Inc. Document Number 007-2590-001, *Developer Magic's RapidApp User Guide*, 1994.

[Tolliver 96] Tolliver, John V., *Studies on Submarine Control For Periscope Depth Operations*, Master's Thesis, Naval Postgraduate School, Monterey California, September 1996.

[UG-IV.9] Naval Postgraduate School, NPSNET IV.9 Documentation, *User's Guide*, Monterey California, 1996. Available at *http://www-npsnet.cs.nps.navy.mil/npsnet/distrubition/doc/SystemOverview_1.html*

[Stewart 96] Stewart, Bryan C., *Mounting Human Entities to Control and Interact With Networked Ship Entities in a Virtual Environment*, Master's Thesis, Naval Postgraduate School, Monterey California, March 1996. Available at *http://www-npsnet.cs.nps.navy.mil/npsnet/publications.html*

[VESUB 97] Naval Air Warfare Training Systems Division, *VESUB Technical Description*, Orlando Florida, 1997. Web page accessed 2 March 1997, available at *http://www.ntsc.navy.mil/tech/vesub/industry.html*

[Zyda 94] Zyda, Michael J., Pratt, David R., Kelleher, Kristen M., *1994 Annual Report for the NPSNET Research Group*, Naval Postgraduate School, Monterey California 1994. Available at *http://www-npsnet.cs.nps.navy.mil/npsnet/publications.html*

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, Ste 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chair, Code CS 2
   Computer Science Department
   Naval Postgraduate School
   Monterey, CA 93943-5000

4. Dr. Michael J. Zyda, Professor 2
   Computer Science Department Code CS/ZK
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Dr. Don Brutzman, Assistant Professor 2
   Undersea Warfare Academic Group Code UW/BZ
   Naval Postgraduate School
   Monterey, CA 93943-5000

6. John S. Falby, Senior Lecturer 2
   Computer Science Department Code CS/FA
   Naval Postgraduate School
   Monterey, CA 93943-5000

7. Dr. Robert Hays 1
   NAWCTSD 4926
   12350 Research Parkway
   Orlando, FL 32765

8. LT Jeffrey A. Halvorson 2
   1134 Hayward Ave.
   Bremerton, WA 98310